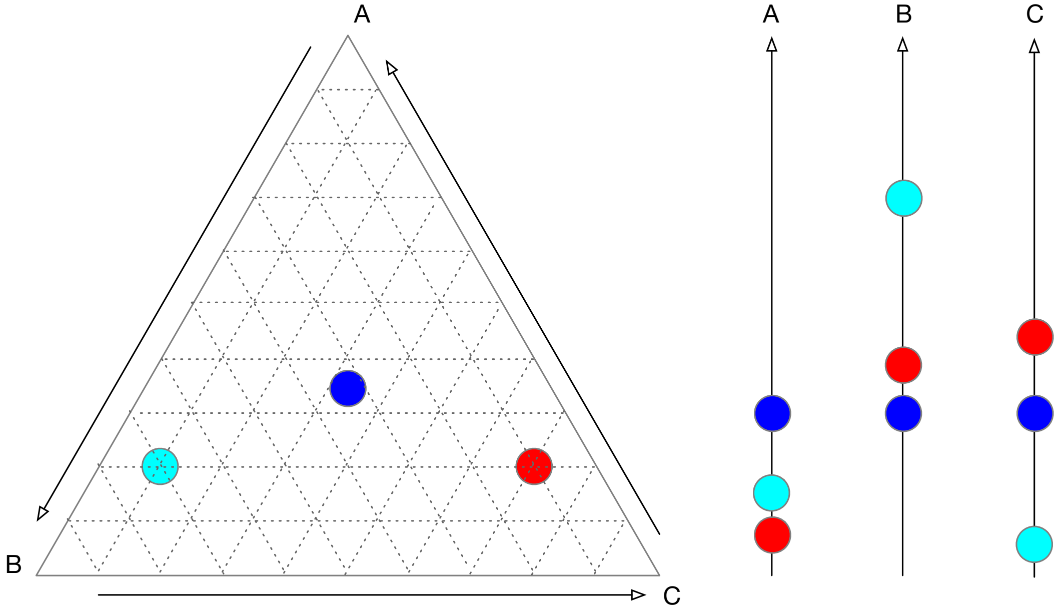
Contents

```
title: “Compositional analysis of high throughput sequencing data”
author: “Greg Gloor”
date: “24 November, 2021”
documentclass: book
csl: [/Users/ggloor/Documents/0_git/csl_styles/apa.csl]
link-citations: yes
output:
bookdown::pdf_book:
keep_tex: true
number_sections: FALSE
toc_depth: 3
includes:
in_header: /Users/ggloor/Documents/0_git/templates/header.tex
pandoc_args: [
“-V”,
“classoption=onecolumn”,
```

“–bibliography=/Users/ggloor/Library/texmf/bibtex/bib/bibdesk_refs.bib”
]

Preface



Why this book exists

This is an attempt to put on (virtual) paper my thoughts on the use of compositional data analysis methods to examine high throughput DNA sequencing experiments. Primarily it is intended as a guide to graduate courses and as an aid when presenting workshops.

Structure of the book

The first half of the book is some theory and justifications. This is written assuming the reader is a graduate student in a biological or biomedical science and that the student has little background in statistics or bioinformatics.

Information and conventions

This document is an .Rmd document and can be found at:

github.com/ggloor/book

The generation of this document requires `R` and an installation of LaTeXto work properly. This document contains interspersed markdown and `R` code that may be compiled into a pdf document and supports the figures and assertions in the main article. `R` code is (almost always) exposed in the pdf document so that the interested reader can work through the example code themselves. Code that is not exposed is in the `chunk` directory.

Reproducing the analysis

From an R command prompt you can compile this document into PDF if you have LaTeXand pandoc installed:

`bookdown::render_book("index.Rmd")` or you can open the file in RStudio and compile in that environment.

R packages required

We will need the following R packages, and in addition several functions are defined in a dedicated functions section.

1. knitr (CRAN)
2. bookdown (CRAN)
3. vegan (CRAN)
4. ALDEx2 (Bioconductor)
5. propr (CRAN)
6. Ternary (CRAN)

About the author

Oh god I'm old and boring

Introduction

> "What really is the point of trying to teach anything to anybody?"
>
> This question seemed to provoke a murmur of sympathetic approval from up and down the table.
>
> Richard continued, "What I mean is that if you really want to understand something, the best way is to try and explain it to someone else. That forces you to sort it out in your mind. And the more slow and dim-witted your pupil, the more you have to break things down into more and more simple ideas. And that's really the essence of programming. By the time you've sorted out a complicated idea into little steps that even a stupid machine can deal with, you've learned something about it yourself. The teacher usually learns more than the pupils. Isn't that true?" [Richard MacDuff] [a]

> Let us think the unthinkable, let us do the undoable, let us prepare to grapple with the ineffable itself, and see if we may not eff it after all. [Dirk Gently] [b]

[a] Douglas Adams, 1987 in *Dirk Gently's Holistic Detection Agency*, William Heinemann Ltd, London

[b] ibid

This booklet is intended for use in teaching graduate student courses and conference workshops on using compositional data analysis methods to examine high throughput sequencing datasets. The approach taken here is largely intuitive and hands on. Formulas for basic methodologies are presented, but the intuitive reason for using them takes precedence. The methods presented here have been used for 16S rRNA gene sequencing, transcriptomics, metagenomics and in-vitro selection (selex) experiments.

The first section is background and theory using toy examples. The second section is application of what we have learned using practical examples. I hope you find this useful.

Outline of the material

- I begin with a brief overview of sequencing technologies, an overview of the types of data we are likely to encounter, and describe how and why these instruments generate data that are constrained to a constant count.
- I introduce sequencing as a stochastic process, explain why we need to estimate our technical variance and show how this can be done technical variance
- I next semi-formally introduce compositional data, and show with examples the pathologies associated with this type of data.
- I then introduce common data transforms and distances used in the high throughput sequencing literature, and demonstrate that none of the transforms affects the compo-

sitional nature of the data, and that in fact, many of the transforms affect the data in non-intuitive ways

- I begin the practical part with exploratory data analysis using the compositional biplot
- I describe the properties of three types of plots to examine high dimensional data: Bland-Altman plots (MA plots), volcano plots, and effect plots
- I describe compositionally appropriate methods to estimate differential abundance with an emphasis on ALDEx2 and to a lesser extent ANCOM
- I describe compositional association as a replacement for correlation using the propr R package

The nature of sequencing data

There are a tremendous number of high throughput sequence analysis tools in the literature. The vast majority of these are recommended for use in only one domain. Domain specific tools are found in all experimental designs and are often touted as 'optimized for' a particular design. Another, less charitable way of describing a domain-specific tool is 'over-parameterized'. It is important to make as few assumptions as possible when examining data, and to ensure that the data being analyzed and the assumptions of the analysis tools are met [@box:1976].

> Since all models are wrong the scientist cannot obtain a "correct" one by excessive elaboration. On the contrary following William of Occam he should seek an economical description of natural phenomena. Just as the ability to devise simple but evocative models is the signature of the great scientist so overelaboration and overparameterization is often the mark of mediocrity.[a]

[a] George Box 1976 *Science and Statistics.* J. Am. Stat. Soc. 71:791

A constrained random sample

All high throughput gene sequencing datasets share a common origin and it is important to understand the source of the data. In essence, an arbitrarily large number of DNA molecules is randomly sampled from an environmental population of molecules and a small, fixed number of those molecules are sequenced.

There is a frequent assertion that data generated by high throughput sequencing instruments are counts. On the surface, this makes sense because we map reads to intervals and we observe the number of counts per interval. However, problems arise immediately upon reflection. One pervasive issue is that the results are strongly influenced by the total read count per sample, and this is the primary reason for almost all tools and workflows using 'count normalization' methods that are introduced later.

The 'Open Random Sample' panel in Figure 1 shows how random sampling by sequencing is usually thought of. We start with an imaginary population containing four different randomly distributed entities, tigers, ladybugs, aliens and rabbits. We want to infer the abundance of each entity in the population by taking a random sample. This is done by choosing a particular area (or volume) to sample and counting the number of each entity in the area. Doing this, we observe 14 rabbits, 1 alien, 24 ladybugs and 6 tigers. We can use this random sample to infer something about the abundances of the entities in the entire population if we know the size of the sampled area and the total size of the environmental area.

A single random sample will of course limit the precision of estimation of the population characteristics. The reason that we replicate experimental measurements is to estimate and to place limits on the precision of the estimates of population parameters. A second random sample is shown on the right and is labeled as a 'closed' random sample (the constraint will be explained shortly). The second random sample, with the same area contains 12 rabbits, one alien, 26 ladybugs and 4 tigers. The first estimate had a total of 45 entities, and the second had a total of 43 entities. Two important points need to be made. First, we

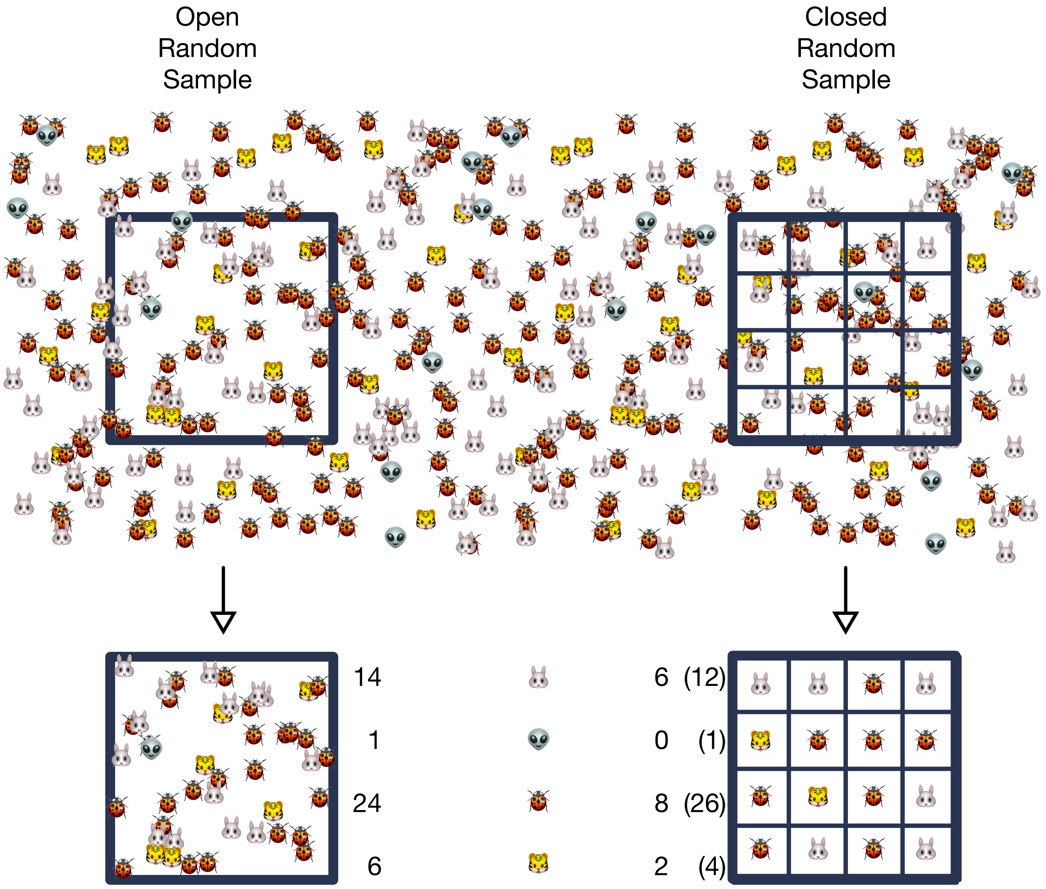


Figure 1: DNA sequencing is a defined-limit random sample of the environment. We start with a population of four entities randomly distributed; rabbits, aliens, ladybugs and tigers. The standard approach is to randomly sample a fixed area of the field. This is an 'open' random sample since the *number* of entities found inside the sampling area is a direct readout of the sampled area. Further, the number of one entity in the area is generally not correlated with the number of the other entities. In contrast, DNA sequencing is constrained by the number of fragments that the machine can accomodate. This is akin to filling the squares on a checker board, where only one entity is allowed per square. This is shown in the 'closed' random sample example.

are comparing the numbers of each type of entity, and second, the total number of entities identified in each sample is free to vary. In theory, we could sample the same area that is very densely populated by ladybugs (perhaps it is mating season) and identify thousands of ladybugs in a very small area.

DNA sequencing is not a counting exercise because the total number of entities found is constrained by the capacity of the instrument. Imagine the worlds worst high throughput sequencer, the GS16™ (GregSeq16™). The GS16™ has a total capacity of 16 molecules, any more and it fails, any fewer and the customer is upset because they did not receive the number of reads promised by the very slick sales brochure. The process of filling the 16 slots proceeds by a random process whereby the first entity to occupy a slot is sequenced, and any subsequent entities are unable to bind to the proprietary surface.This is akin to the processes involved in all current high throughput sequencing instruments. Applying this random process to the entities in the second sample, we observe that the GS16™ delivers 6 rabbits, 0 aliens, 8 ladybugs and 4 tigers. In the event that we sample from the area where the ladybugs were mating, we would recognize only that there were 16 ladybugs in the area, and miss entirely that the area was a seething mass of ladybugs: that is, we would be unable to distinguish if the sample just happened to be from a site that was missing the other entities, or from a site that contained a vast number of ladybugs.

Constrained and un-constrained environments.

If we were sampling from a closed system, the closed random sample approach might be an acceptable approximation of the underlying environment. However, I argue that most environments are not closed but open, and these will be referred to as constrained and unconstrained. A constrained environment is operating at some pre-determined limit or carrying capacity, and in constrast in an unconstrained environment the total number of entities in the environment is free to change.

In Figure 1 we were modelling what happens when the environment is un-constrained, that is, the environment can contain arbitrarily greater or fewer numbers of any entity. Two real-world examples spring to mind of such an un-constrained environment. The first example would be the consortium of organisms that inhabit the human vagina and which occur in at least two major states. The first state is dominated by one or a small number of species from the genus *Lactobacillus*. The second state is dominated by a mixed population of anaerobic species. The total numbers of bacteria (bacterial load) in the second state is about 100-fold greater than in the second state [@Zozaya:2010]. When comparing either DNA or RNA molecules from these two populations we would need to ensure that we account for the un-constrained nature of the system and not assume that the total number of molecules sampled is equivalent. The second example of an un-constrained environment would be in-vitro selection experiments as published in [@mcmurrough:2014;@Wolfs:2016aa] where there are two different classes of genes; inactive genes where the absolute abundance of the corresponding DNA molecules remains unchanged over time, and active genes where the absolute abundance of the corresponding DNA molecules increases exponentially. The third example occurs when comparing normal cells and cells transformed to a cancerous phentoype by expression of an activated cMyc protein. In the cancer case, the total number of RNA molecules is several fold higher than in the matched normal tissue control cells [@Mortazavi:2008]. In all three of these types of experiment, the absolute number of molecules is free to change.

A constrained environment arises when the underlying system is operating at some fixed limit. As an example, a culture of *Escherichia coli*, a common lab strain of bacteria, that is growing in the lab under controlled conditions has a near constant number of RNA molecules in each cell when grown to a predetermined cell density [@mRNA:2002]. Moreover, when a new gene is induced to make new mRNA molecules under these standard conditions the number of RNA molecules of very highly expressed genes is reduced to compensate [@Taniguchi:2010aa]. Thus, under a given condition, the total number of mRNA molecules in a cell has a fixed limit: the cell cannot exceed this limit under the given condition. In this case all mRNA molecules in the system are inherently coupled, a change in absolute abundance of one molecule is compensated for by a change in absolute abundance by one or more others to ensure that the total number of molecules in the system is relatively constant. In the language of sports analogies "a single cell cannot give 110% effort". It is assumed that mRNA molecular abundance in most lab cultures of most cell types behave similarly.

Obviously many real datasets exist on a continuum between these example extremes, but it is not always obvious at which extreme a given sample lay. For example, when comparing gene expression in liver and kidney cells do we expect the same number of underlying molecules per cell? What about comparing liver and red blood cells? Here the answer is more obvious since the red blood cells are much less metabolically active and express far fewer unique mRNA molecules than does the typical liver cell. Knowing if the samples are from a constrained or un-constrained system has important implications for analysis as outlined below. As we shall see in the practical examples, most tools work acceptably well with constrained data, but fail in unexpected ways on un-constrained data.

Modeling constrained and un-constrained samples

A simple thought experiment should clarify the importance of knowing the difference between constrained and un-constrained systems. In this example I generate a test dataset composed of 100 features in 20 samples—modelled as a time series with 20 steps. An equally valid way of thinking about this would be if one feature had a growth advantage over the others. There are two situations. In the 'constrained' situation, one feature increase exponentially in each time step and one other feature decreases to compensate. In the 'un-constrained' situation, only one feature increases while the remainder of the features remain at a constant total abundance.

```
num.one = 90 # base number of features

m.dub <- prop.m <- clr.m <- m.dub.u <- prop.m.u <- clr.m.u <-
    matrix(data=NA, nrow=20, ncol=num.one + 10)

in.put <- c(10,10000,1,5,10,20,50,100,200,1000) # arbitrary data

# ensure feature 3 is arbitrarily large and get the constrained total
```

```
total.sum <- sum(in.put + 1) * 1000

# one feature increases exponentially and one feature decreases to compensate
for(i in 0:19){
    # un-constrained data, feature 1 exponentially increases
    m.dub[i+1,] <- m.dub.u[i+1,] <- in.put * c(2^i, rep(1,num.one + 9))
    prop.m.u[i+1,] <- m.dub.u[i+1,]/sum(m.dub.u[i+1,])
    clr.m.u[i+1,] <- 2^(log2(prop.m.u[i+1,]) - mean(log2(prop.m.u[i+1,])))
    # now constrain the data, feature 3 decreases to compensate
    m.dub[i+1,3] <- total.sum - sum(m.dub[i+1,])
    prop.m[i+1,] <- m.dub[i+1,] / sum(m.dub[i+1,])
    clr.m[i+1,] <- log2(prop.m[i+1,]) - mean(log2(prop.m[i+1,]))
}
```

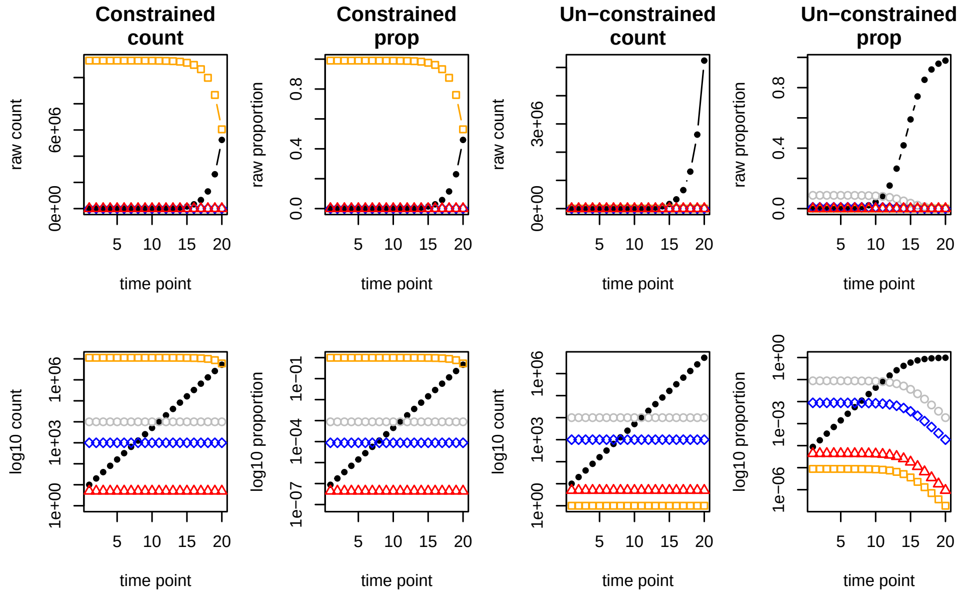


Figure 2: Constrained and un-constrained data are very different. The 'Constrained count' column shows a synthetic dataset where the total number of counts is a constant and this is plotted as counts on top and on a log10 scale on the bottom. Here we see that the orange and black features are compensating and perfectly negatively correlated. These same data are next converted to proportions or relative abundances and plotted in the 'Constrained prop' column. Both the count and proportional data are the same, except for the scale, when the data are constrained. The un-constrained data behave differently. The 'Un-constrained count' and 'Un-constrained prop' columns are distinct and the proportional data are severely distorted. The un-constrained counts are all independent, but the exponentially increasing feature in black is negatively correlated with the constant features in the un-constrained proportion plot. Only 5 of the 100 features are shown for clarity.

Of course DNA sequencing is not the same as our synthetic example. When collecting samples and sequencing them, there are many more features, and significantly more noise because of random sampling than is modelled in Figure 2. There are many sources of sampling error or even sampling bias. Nevertheless, this example serves as an instructive starting point.

Instrument capacity

So how is sequencing a constrained operation? Each instrument has its own specific capacity issues that must be taken into account.

The Ion Torrent and Ion Proton systems have a chip with a predetermined number of pores on the sequencing chip (10 of thousands to 10s of millions, depending on the chip) that can accept an amplified library fragment. No signal is returned from an empty pore, and the signal is rejected if a pore contains two or more different fragments. Sequencing is successful only when the pore is occupied by a single fragment from the library. This is directly analogous to the Closed Random Sample panel in Figure 1

The Illumina sequencing instruments attach the DNA fragments to a glass slide and then each fragment is amplified into millions of identical fragments called clusters which appear as randomly distributed spots under a microscope. Each different Illumina instrument accommodates a characteristic maximum number of clusters, and if two or more clusters overlap they are rejected by the software. The newest Illumina instrument, the NovaSeq, has a fixed number of cells. Thus, there is a fixed number of spots that can be accomodated on the Illumina sequencing chip just as there are a fixed number of slots on the Ion platforms or the GregSeq.

Therefore, regardless of instrument, the technician must apply a precise number of DNA molecules that maximizes the number of fragments on the sequencing instrument without overloading it. It should be obvious that loading a DNA sequencer is akin to filling the squares on a checkerboard where the goal is to have as many checkers as possible, without overlapping the pieces. DNA sequencing instruments have an upper bound on the number of fragments they can sequence, and as we shall see later, any arbitrary upper bound is equivalent to a proportion. This means that high-throughput sequencing affects the shape of the data differently on constrained and un-constrained data as shown on Figure 2.

It is often assumed that the abundance of each input DNA species that is observed after sequencing reflects in some linear way a random sample of the input molecules. This is likely to be the case if the total number of molecules in the input sample is constrained. Such a constraint would be met if, for example, an increase in one or more DNA species was balanced with an equivalent decrease in one or more different species. However, as we shall see, the analysis of un-constrained data will be a problem if the limited output of the sequencing platform is not accounted for.

Example calculation of fragment number

The number of fragments after sequencing is determined by the instrument; an Ilumina MiSeq delivers $\sim$ 20M fragments whereas an Illumina NextSeq delivers $\sim$ 400M and an

Illumina HiSeq can deliver $\sim$ 250M reads per lane on each of 8 lanes. The commonly used Nextera DNA library kit is optimized to require 50 ng of DNA per sample. Thus, the number of fragments of DNA (or RNA) molecules in the underlying environment, in general, vastly outnumbers the number of sequence fragments from which the library is made, and the number of fragments in the library in turn outnumbers the number of fragments from which sequencing data are ultimately derived. We can do a simple back of the envelope calculation for an example metagenomics sequencing run to show this.

Assume that we have a mixture of bacterial species with a mean genome size of 4 Mb. One mole of genomes would have a mass of 2.64×10^9 grams. A typical environmental bacterial density when collecting a metagenome sample would be on the order of at least 10^7 bacteria per ml of sample, so if we isolate the bacteria occurring in a 1 ml sample, this corresponds to 10^7 genomes, which corresponds to about 44 ng of DNA.

If the DNA concentration after isolation is 1 ng/μl, and one μl of DNA is taken, this corresponds to $(1 \times 10^{-9}$ g$)/(2.64 \times 10^9$ g/mole$) \times (6.02 \times 10^{23}$ genomes/mole$) = 228,000$ genomes. The Illumina Nextera XT kit can be used to make a library with this amount of DNA. Recall that the the DNA is fragmented, typically into 500 bp or smaller sizes. Using a fragment size of 500 bp, this corresponds to approximately 1.8×10^9 DNA fragments even for a measly 1ng of input DNA.

In the scenario where a single sample is prepared and the maximum number of fragments are loaded and run, this still results in only 1% of DNA fragments in the library being sequenced on the Illumina MiSeq, and only 22% of the DNA fragmments being sequenced on the Illumina NextSeq.

An even smaller proportion of each sample is typically sequenced. DNA sequencing rarely involves a single sample, but instead samples are 'multiplexed' on the sequencing run by mixing two or more libraries together. When this occurs the samples have a unique tag, or barcode, attached so that the samples can be uniquely identified post sequencing [@Parameswaran:2007aa;@Andersson:2008;@Gloor:2010]. Barcodes can be added by ligation or by incorporation into PCR primers that are used to amplify the library. Obviously, increasing the number of samples through multiplexing will result in an even smaller proportion of the fragments in each sample being sequenced.

Sequencing post processing

After sequencing fragments are grouped in some way

Mapping

Fragments generated from metagenomic or RNA-seq experiments are aligned to reference sequences corresponding to genes, transcripts or genetic intervals (generically genes). The total number of fragments mapping to a given gene is said to be the read count for that gene and the read count for all genes in a system ranges from 0 (no fragments align to that gene) to the total number of fragments in the sample. The output from these types of experiments is a table of read counts per gene per sample. These genes can be further aggregated into functional categories.

See later chapters for example workflows.

OTU generation

Fragments generated from tag-sequencing are often merged into operational taxonomic units (OTUs) at some predefined percent identity, or tabulated by the number of identical fragments observed (ASUs). The output from these types of experiments is a table of counts per OTU per sample.

Random sampling by sequencing

A counting operation always allows the addition of 'one more observation', and is one of fundamental operations that integer mathematics and statistics depend upon [@number]. In the context of DNA sequencing this would equate to loading the samples onto an Illumina MiSeq chip to the optimal fragment density and then adding in 'one more fragment' repeatedly until the number of fragments loaded was equivalent to that on a much higher capacity chip (say the NextSeq). However, this is not possible since at some point the total number of fragments would exceed the capacity of the chip and the sequencing reaction will fail. Stated bluntly, one cannot purchase an Illumina MiSeq run and expect to receive data equivalent to an Illumina NextSeq or HiSeq run. This self-evident fact is completely overlooked in the literature.

In Chapter we saw that DNA sequencing is not a counting operation but is instead a random sampling operation with a fixed sum from a large pool of molecules. This is akin to sampling different colored balls from an urn that contains more balls than are sampled and stopping after a fixed number of balls are drawn.

The random sampling can be direct at the level of sampling from the environment. In the case of genomics or metagenomics DNA is made directly from the sample. Sampling can be indirect in the case of RNA-seq where RNA molecules are sampled after they have been converted to DNA via reverse transcription. Note that an environment may not be uniform. For example, it has been observed that different microbial compositions are observed when collecting stool samples if a sample is taken from the interior or exterior of the stool [@Gorzelak:2015aa]. Sampling can also be indirect in the case of tag-sequencing or single cell sequencing. In tag-sequencing, most typically applied to 16S rRNA gene sequencing, a small defined region is amplified by the PCR, and the amplimer size is usually compatible with the sequencing instrument. In single cell sequencing the molecules from a single cell are fragmented and amplified as a mixture.

Fragment sampling occurs because as noted above the fragments actually sequenced are a random sample of the fragments that are in the sequencing library. There are always more fragments in the library than can be accommodated on the instrument, and there are almost always more molecules in the environment than can be accommodated in the library. The sole exception to this rule would be when sequencing is used to investigate low biomass environments—however in this case the library protocols always have an amplification step that increases the number of fragments above the number that can be accommodated on the instrument. After the DNA is made and fragmented, an aliquot of the population of DNA fragments are used to make a *sequencing library* by attaching standard sequences that permit the DNA fragments to be bound to the solid phase of the sequencing chip that is particular to a given platform. Fragmentation is not typically employed in tag-sequencing since the amplified DNA fragments are typically small.

Finally, a sample of the library is loaded onto a sequencing platform, and this sampling occurs by a multivariate Poisson process [@fernandes:2013] as outlined below. At this point two or more independent libraries are usually mixed together into a multiplex library, and this adds a third level of randomness into the process. The number of fragments in each library is one of the strongest influences on the apparent information in the sample

[@Horner-Devine:2004aa;@Weiss:2017aa]. In other words, the number of fragments identified in a sample post sequencing is a confounding variable. The number of fragments observed post sequencing is termed the 'library size' or the 'depth of coverage'. It is important to remember, again, as outlined in Chapter that all sequencing platforms in widespread use contain a fixed number of locations to which the DNA fragments can bind.

If the number of DNA fragments sequenced has an arbitrary upper bound determined by the machine, and the number of fragments in the library is always larger than the machine capacity, then it should be obvious that the number of fragments sequenced can contain no information about the *number* of fragments in the library pool, nor can the number of fragments contain information about the *number* of molecules in the original DNA sample from the environment. The univariate logical equivalent is to only know the percentage that a suit is marked down to, without knowing the original price: the customer would have no idea whatsoever about how much money it costs, only that they were getting a helluva deal. The multivariate intuition is that we cannot know the number of balls of different colour in the urn, we can only infer their proportions.

Looking back at the GregSeq example in Figure 1, if we always recover 16 entities, and all of them are ladybugs, we cannot know if the sample was from a high or low density of ladybugs. All we can say is that we only recovered ladybugs. Therefore, the only information available is the *relative* proportion of individual fragments in the library, which is assumed to approximate the relative proportion of fragments in the DNA sample from the environment. We will revisit this issue when we discuss normalizations in common use.

Probabilistic sampling

It is simple to demonstrate that the DNA sequencing results obtained is a probabilistic and not a deterministic sampling process. In the former case we should expect some variation due to the act of sampling, in the latter case we expect the same result every time. Fortunately there is a good system to test this by examining reads from adjacent lanes in the same flow cell. For this analysis we use a well-controlled dataset of *Saccharomyces cerevisia* RNA-seq experiment carried out with seven technical replicates per sample [Gierlinski:2015aa,@Schurch:2016aa]. We take the dataset, which has 7 technical replicates of each biological sample and compare a reference technical replicate with the other replicates. We will examine only the first set of technical replicates, but the concepts hold for the other 95 technical replicates in the dataset and for other similar datasets, e.g. in [@fernandes:2013].

Figure 3 shows that the observation of a count in one technical replicate is a good, but not perfect predictor of the count in a second technical replicate. We observe some random variation around the line of equality. There is also a systematic difference between the expected value near the low count margin and the expected value at higher counts. At the low count margin the expected value across multiple technical replicates is greater than the count observed in the reference replicate. The ratio panel on the right shows that this holds true until the reference replicate has reached 10-15 counts—this depends upon the read depth. Conversely, the reference and the expected value are close matches at higher counts, although there appears to be a slight tendency for the reference to over-estimate the expected value. Thus the low count margin is over-dispersed, leading to the usage of negative binomial methods, or zero inflated Gaussian methods to model the data. However, we have found that a Dirichlet distribution with a uniform prior of 0.5 is a good fit to the data [@fernandes:2013; @gloorAJS:2016].

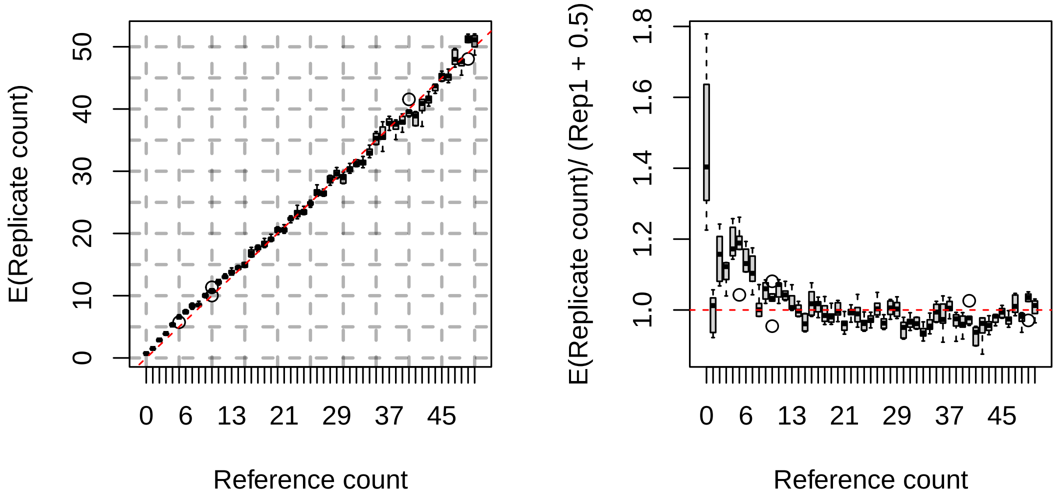


Figure 3: One technical replicate was taken as the reference, and for all genes of a given count on the x-axis, the mean (expected) value of the genes in the other six replicates were determined. The results for the genes with a count in the reference replicate up to 50 are shown in the left panel. The red line shows equality between the reference replicate and all others. The right side panel shows the ratio between the observed value in the reference replicate after the addition of a prior expected value of 0.5, and the expected values.

A Formal Description

Definitions and notation

1. sample vector $\vec{\mathbf{s}}_i$
2. samples $i = 1 \ldots n$
3. feature vector $\vec{\mathbf{f}}_j$
4. features or parts $j = 1 \ldots D$
5. feature value $\mathbf{s}_{ij}$
6. the environment Ω
7. sample geometric means $g_i = (\prod_1^D s_i)^{1/D}$
8. random instances of the data $k = 1 \ldots m$

It is worth recalling that essentially all HTS data come from underpowered experimental designs, in the sense that there are more features than there are samples. Thus, the strength of evidence for statistical inference must be weak [@Halsey:2015aa]. Paradoxically, the features that are identified as differentially abundant must *appear to be very different*, much more so than the actual data support [@Colquhoun:2014aa;@Halsey:2015aa]. The combination of small sample sizes, large numbers of variables and the mis-use of the null-hypothesis testing framework are a deadly combination [@forking:2013].

Any results can only be validated by independent replication, meta-analysis assuming all experiments are published, or by an orthogonal method [@Cumming:2008aa]. All of which are rare in the transcriptome, metagenome, and microbiome fields.

When estimating differential abundance it is important to properly estimate the dispersion, τ, of the j^{th} feature for all samples; dispersion of a feature can be represented by the following simple model:

$$\tau_j = \nu_j + \epsilon_j \tag{1}$$

where ν represents the underlying biological variation and ϵ represents the stochastic error from all the steps involved in the collection, preparation, and sequencing of the dataset outlined above. For a given experiment, we *only* have the biological variation available from the biological samples themselves. There is no principled way that additional biological samples can be imputed and so ν is fixed by the sample size.

However, all steps in the collection and sequencing involve random sampling from a larger pool of molecules, and can be approximated by a model of drawing different colors of balls from an urn that contains many more balls than are drawn. Under this assumption, repeated sampling of each feature would be expected to be distributed according to a Poisson distribution. Given that the samples are multivariate, we expect a multivariate Poisson sampling process to be appropriate, and this is equivalent to sampling from a Dirichlet distribution with a uniform prior [@Jaynes:2003; @fernandes:2013]. Thus, while ν is fixed by the sample size, in principle and in practice, we can infer ϵ. Under this analytic process we are able to identify those features that have a biological difference that is robust to simple random sampling as outlined below.

The majority of extant analysis tools utilize point estimates of both parameters and there are several underlying similarities in the models used. First, it is generally assumed that ϵ is small relative to ν. Second, it is assumed that there is some underlying similarity in the distribution of ν and ϵ for all features in all samples at a given relative abundance level. That is, if the n features were ordered by abundance, that the expected value of ν_j would be approximately

$$\sum \nu_{j-m} \ldots \nu_{j+m}/2m \tag{2}$$

where m is some small offset in the abundance index. Similar logic applies to estimating the expected value of ϵ, but many tools offer more complex additional models to estimate these parameters for troublesome data. Third, the data are observed to be over-dispersed; that is, the data are observed to have a greater variance than expected from Poisson sampling alone. As we saw in Figure 3 see random sampling alone can account for the apparent 'overdispersion' at the low count margin.

The low count overdispersion has led many tools to model the dispersion using a negative-binomial model, where the dispersion can be greater than the mean. The negative-binomial model is very attractive and widely used in both transcriptome and microbiome studies [@Gierlinski:2015aa;@McMurdie:2014a;@Kvam:2012;@Robinson:2010]. Fortunately negative binomial based models work well when the samples are collected from environments near the constrained end of the spectrum. Unfortunately, these models do not fare as well at the un-constrained end of the spectrum [@gloorAJS:2016;@fernandes:2014;@macklaim:2013;@fernandes:2013], and even worse, tools based on these models rarely fail gracefully with a helpful error.

When absolute variance is measured and plotted vs. sample count, the variance approximates the count as shown by the dotted grey line of equivalence. This is what is expected from a multivariate Poisson process. However, as we shall see in Chapter , actual variance is usually somewhat greater than the mean in real datasets.

When the variance of the ratio of the random samples to the actual data is plotted vs. the sample count the relationship between variance and sample count is exactly reversed. Here the relative variance is greatest at the low count margin and least at the high count margin. We will return to this point in Chapter , but at this point, the reader needs to know that HTS data are *relative and ratio* data by construction.

```
n.sam <- 50 # samples
max.num <- 100000 # max value
# semi-random vector
z <- floor(runif(n.sam, 1, max.num)) # get random integer values
z[1:2] <- c(1,2) # ensure always 1 and 2 values
z[3:10] <- floor(runif(8,3,20)) # some small values
z[11:20] <- floor(runif(10,21,50)) # some intermediate values

# samples by row, features, by column
# random counts from multivariate Poisson
z.dir <- rdirichlet(n.sam, z) * sum(z)
abs.var <- apply(z.dir, 2, var)

# relative counts
z.r <- t(apply(z.dir, 1, function(x) x / z) )

par(mfrow=c(1,2))
plot(log10(z), apply(z.dir, 2, var), main="Absolute variance", log = "y",
    ylab="Variance", xlab="log10(count)")
abline(0,1, lty=2, col="grey")
plot(log10(z), apply(z.r, 2, var), main="Relative variance", log="y",
    ylab="Variance", xlab="log10(count)")
abline(0,-1, lty=2, col="grey")
```

We observed that ϵ can be exponentially larger than ν at the low count margin when measured on a relative scale [@fernandes:2013;@gloorAJS:2016], and that properly accounting for this realization alone can result in an excellent fit to even problematic data. Thus, a reliable analysis can be obtained by incorporating an ‘in silico’ technical replication which explicitly models the variation in ϵ as a probability density function on a per feature, per

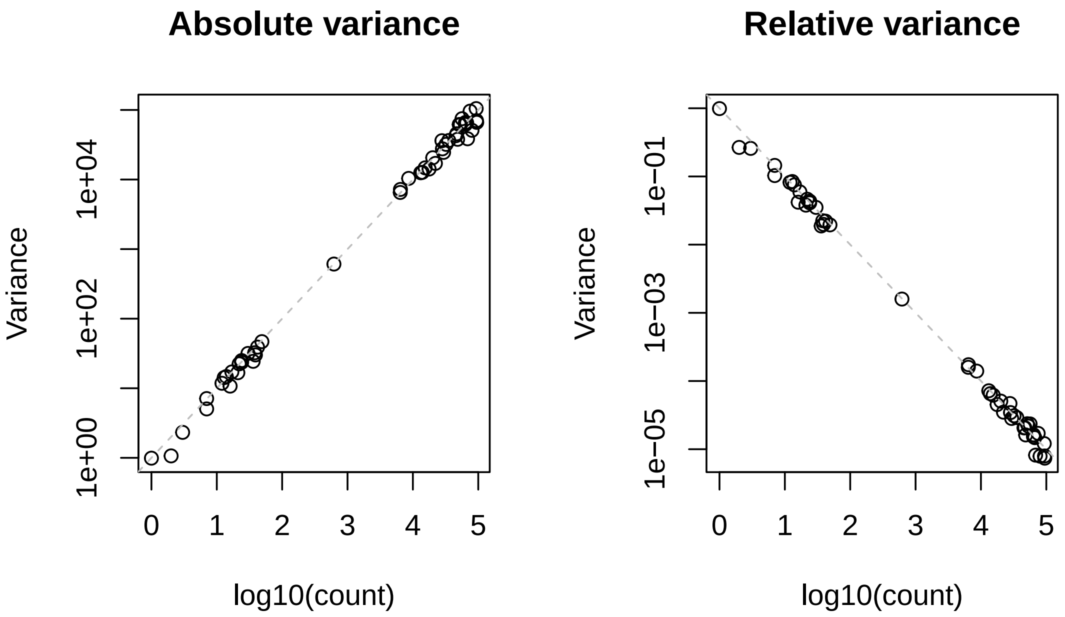


Figure 4: Variance in constrained data is not what we expect. Numbers between 1 and 1000 were generated (the sample) and converted to 50 random instances using a multivariate Poisson process by sampling from the Dirichlet distribution (the instances). The absolute variance of 50 random instances was determined and plotted as the absolute variance of the instances vs. the sample count value. The data was transformed by converting each count to the ratio of the instance count to the sample count (relative value) and the variance of these relative values were plotted. The relative variance is greatest at the low count margin and smallest at the high count margin as is observed for actual sequencing data [@fernandes:2013;@gloorAJS:2016].

sample basis; in other words that $\tau_j = \nu_j + f(\epsilon_j)$. This approach is implemented in the ALDEx2 Bioconductor package and substantially reduces the false positive identification rate in microbiome and transcriptome data while maintaining an acceptable true positive identification rate [@Thorsen:2016aa].

The differences between groups, dispersion within groups and relative abundance were calculated using the ALDEx2 R package that uses Bayesian modelling that generates a probability function for ϵ_j that can be used to place bounds on the uncertainty of the the observed data [@fernandes:2013;@gloorAJS:2016]. If there are two groups, A and B, this requires that the data comparison is properly centred on the difference between these groups. ALDEx2 has been shown to give meaningful and reproducible results, even on sparse, asymmetric datasets using many different experimental designs [@fernandes:2013;@macklaim:2013;@fernandes:2014;@mcmurrough:2014], although as shown here the asymmetry can still affect the outcome.

The starting point for analysis is an n samples $\times D$ features array. The sample vector contains the number of reads mapped to any of the j features in the i^{th} sample, $\mathbf{s}_i = [j_1, j_2 \ldots j_D]$, where $i = 1 \ldots n, j = 1 \ldots D$. The total number of counts is irrelevant and determined by the machine [@Gloor:2016cjm;@gloor2016s]. These data are compositional and are an example of an equivalence class with $\alpha_i = \sum \mathbf{s}_i$. In theory, the vector $\boldsymbol{s}_i$ can be adjusted to a unit vector of proportions, $\boldsymbol{p}_i = [p_1, p_2 \ldots p_D]$, i.e. $\alpha = 1$, without loss of information by the maximum likelihood (ML) estimate $\boldsymbol{p}_i = \boldsymbol{s}_i/\alpha_i$. In this representation, the value of the j^{th} feature is a ML estimate of the probability of observing the counts conditioned on the fractional f that the feature represents in the underlying data and on the total read depth for the sample; i.e., $\mathbb{P}_{i,j}(f_{i,j}|\alpha_i)$. However, the maximum likelihood estimate will be exponentially inaccurate when the dataset contains many values near or at the low count margin [@Newey:1994] as is common in sparse HTS data. Instead we use a standard Bayesian approach [@Jaynes:2003] to infer a posterior distribution of the unit vector directly from $\boldsymbol{s}_i$, by drawing k random Monte-Carlo instances from the Dirichlet distribution with a uniform, uninformative prior of 0.5, i.e.:

$$\mathrm{P}_{i(1\ldots k)} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_k \end{pmatrix} = \begin{pmatrix} p_{i,11} & p_{i,21} & p_{i,31} & \cdots & p_{i,D1} \\ p_{i,12} & p_{i,22} & p_{i,32} & \cdots & p_{i,D2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{i,1k} & p_{i,2k} & p_{i,3k} & \cdots & p_{i,Dk} \end{pmatrix} \sim Dirichlet_{(1\ldots k)}(\boldsymbol{s}_i + 0.5) \tag{3}$$

This approach has consistent sampling properties and removes the problem of taking a logarithm of 0 when calculating the CLR because the count 0 values are replaced by positive non-zero values that are consistent with the observed count data [@fernandes:2013;@gloorAJS:2016]. Each of the Monte-Carlo instances, by definition, conserves proportionality and accounts for the fact that there is more information when α_i is large than when it is small. This partially restores scale invariance to the data by providing a distribution of values where the uncertainty of features scales inversely with the read depth [@fernandes:2013;@gloorAJS:2016].

The apparent solution to the sequencing depth problem is to normalize the read count values across samples in some way. One method of normalization to is by subsampling, often termed rarefaction, as this is observed to reduce the influence of sequencing depth on variation in α and β diversity metrics [@Horner-Devine:2004aa;@Weiss:2017aa]. Another is to convert the data to proportions or percentages; these latter values are widely spoken of in the literature as 'relative abundances'. Subsampling is frequently used to estimate the associated sampling error. Some groups have begun advocating the use of normalization methods prevalent in the RNA-seq field [@McMurdie:2014a] but still treat the data as point estimates of the true abundance. There are many other normalizations that are used in the ecological and high throughput sequencing literature and the purpose and effect of these on simulated data are explored in the section on Data Transformations.

DNA sequencing data are compositions

High throughput sequencing generates compositional data

In the Chapter we saw that the capacity of the sequencing instrument imposed an upper bound on the total number of fragments that could be obtained from a given sequencing run. We also saw that the process of sequencing is essentially a random sampling of an environment where the environment contains more fragments than can possibly be sequenced in Chapter . Finally, the data obtained are read counts per genetic interval (gene or OTU) per sample.

The read counts per sample range from 0 to, as a maximum, the total number of reads in the sample. Thus the data are positive integer data with an arbitrary maximum. While the data have an arbitrary maximum, the majority of current tools assume the data are counts and ignore the arbitrary maximum constraint. This assumption is the basis of methods grounded in distributions such as the zero inflated Gaussian (ZIG) [@Paulson:2013aa], negative binomial [@Robinson:2010] and Poisson based models [@auer:2011]. Recent benchmarking has demonstrated that such methods are unpredictable when dealing with highly sparse data [@Thorsen:2016aa], do not control the false discovery rate [@gloorAJS:2016; @hawinkel2017], and behave poorly when un-constrained datasets are examined [@fernandes:2013; @macklaim:2013;@gloorAJS:2016].

Data of this type are called count compositions, and a number of groups have started to work on developing appropriate methods to deal with high throughput datasets as count compostions [@Friedman:2012; @fernandes:2013; @fernandes:2014; @Lovell:2015; @ancom:2015;@Kurtz:2015;@Gloor:2016cjm;@erb:2016;@gloor2016s;@Tsilimigras:2016aa;@Washburne:2017aa;@Quinn:2017;@S

So what is compositional data (CoDa), and what are its properties with respect to high throughput sequencing that make this an important issue?

Compositional data

Data from high throughput sequencing have the following properties; the data are counts, the data are non-negative, and the data has an upper bound imposed by the instrument. This fits with the definition of compositional data: compositional data contain D features (OTUs, genes, etc), where the count of each feature is non-negative, and the sum of the parts is known or arbitrary [@Aitchison:1986, pg25]. Note that the data do not have to sum to a predetermined amount, it is sufficient that the sum of the parts be known and be bounded.

A vector containing D features where the sum is 1 can be formally stated as: $\vec{X} = \{(x_1, x_2, x_3, \dots x_D); x_i \geq 0; \sum_{x=1}^{D} = 1\}$. The sum of the parts is usually set to 1 or 100, but can take any value; i.e., any composition can be scaled to any arbitrary sum such as a ppm. Compositional data are equivalence classes since one vector can be scaled into another through multiplication by an arbitrary constant [@barcelo:2001]. In the lexicon of high throughput sequencing the vector is the sample and the features are the OTUs or genes or genomic intervals. The total sum is the total number of fragments observed for the sample; i.e., the sequencing depth.

CoDa pathologies

Compositional data have a number of built-in pathologies: a negative correlation bias, sub-compositional incoherence, and spurious correlations. A proper analysis of compositional data must as a minimum account for these pathologies.

Formally, compositional datasets have the property that they are described by $D-1$ features [@Aitchison:1986]. In other words, if we know that all features sum to a constant, then the value of any individual feature can be known by subtracting the sum of all other parts from that constant; i.e., $x_D = 1 - \sum_{x=1}^{D-1}$.

Graphically, this means that compositional data inhabit a space called a Simplex that contains 1 fewer dimensions than the number of features. The distances between parts on the Simplex are not linear. This is important because all parametric statistical tests assume that differences between parts are linear (or additive). Thus, while standard tests will produce output, the output will be misleading because distances on the simplex are non-linear and bounded [@martin1998measures]. Chapter on Data Transformations contains an intuitive demonstration of how data are moved to the Simplex when a the data are compositional.

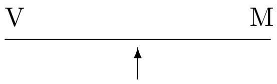
It is not always apparent when the data are compositional. This is especially true for large multivariate datasets such as those generated in high throughput sequencing. Aitchison [-@Aitchison:1986] indicated that a compositionally appropriate analysis should fulfil a number of properties, and when these properties are not met with traditional analyses, the data are likely compositional.

- A compositionally appropriate analysis should be scale invariant, that is, the results should not depend on the total count or scale of the sample. This first principle of CoDa data analysis indicates that all count normalization, or sequencing depth adjustments are either unnecessary or counterproductive. There is substantial resistance to the idea the high throughput sequencing data are compositional, and indeed many analysts believe that the data can be made non-compositional with the 'correct' transform that restores the scale. The is belief is exposed to be false in Chapter .

- A compositionally appropriate analysis should also not depend on the order of the features in the dataset. This almost goes without saying, but is included because of the way that one particular transformation, the alr, was formulated.

- A compositionally appropriate analysis should exhibit subcompositional coherence, or the results of analysis of a sub-composition should be the same as for the entire composition. In practice, this is difficult to achieve, and we settle for least sub-compositional dominance where the distances between features in the full composition are equal to or greater than the distances in the sub-composition. In later chapters where we examine real datasets, I show how to determine if sub-compositional dominance is fulfilled by the analysis.

Negative correlation bias in compositions



The values of the parts of compositional datasets are constrained because of the constant sum, and this constraint has been known for a very long time [@Pearson:1896]. The features in a composition have a negative correlation bias since an increase in the value of one part must be offset by a decrease in value of one or more other parts. In the illustration above, we see that 'V' and 'M' are perfectly balanced on the fulcrum because they have the same mass. If M becomes heavier, then V will rise even though the mass of V has not changed. The same principle operates in compositional data. If V is the amount of money spend on vegetables, and M is the amount of money spent on meat, and the total food budget is a constant, then the only way that more meat could be consumed would be to spend less on vegetables. Therefore, the amount of money spent on V and M will be perfectly negatively correlated if the total food budget is constrained. This example generalizes to any number of items in the shopping basket as long as the total budget is constrained. When there are more items, then an increase in one item (say shoes) must be offset by a decrease in another item, but it could be a decrease in meat, vegetables or both.

Spurious correlations:

In addition to a negative correlation bias, compositional data has the problem of spurious correlation [@Pearson:1896]; in fact spurious correlation was the first troubling issue identified with compositional data. This phenomenon is best illustrated with the following example from Lovell et. al [-@Lovell:2015], where they show how simply dividing two sets of random numbers (say abundances of OTU1 and OTU2), by a third set of random numbers (say abundances of OTU3) results in a strong correlation. Note that this phenomenon depends only on there being a common denominator.

```
set.seed(13)
T <- rnorm(50, mean=100, sd=25)
L <- rnorm(50, mean=10000, sd=2500)
R <- rnorm(50, mean=1000, sd=250)
A <- rnorm(50, mean=5, sd=2.5)
ran.dat <- cbind(T,L,R,A)
ran.dat[ran.dat <=0 ] <- 0.1
```

Sub-compositions

Compositional data have the third property of sub-compositional incoherence of correlation metrics as illustrated in Chapter . That is, *correlations calculated on compositional datasets are unique to the particular dataset chosen* [@Aitchison:1986]. This is problematic because

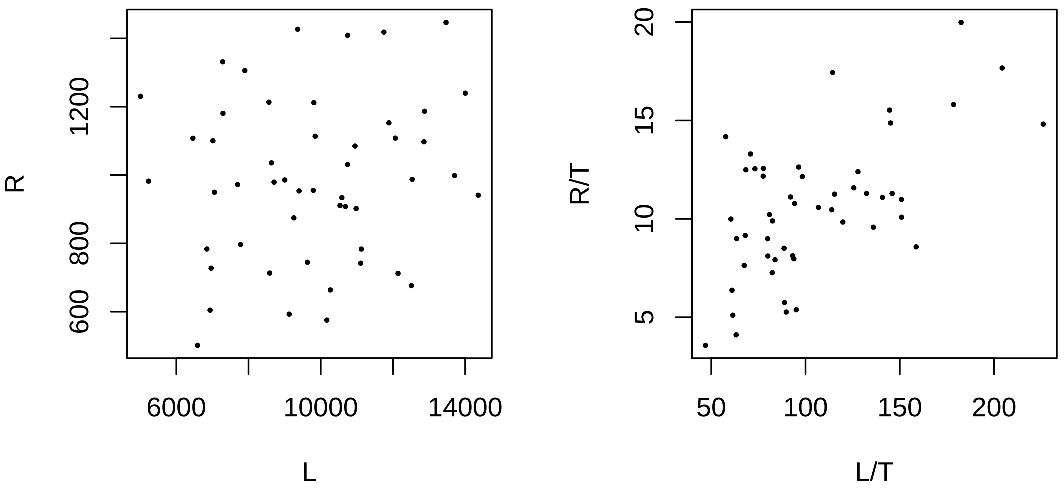


Figure 5: Spurious correlation in compositional data. Two random vectors (R and L) drawn from a Normal distribution as in the above R block, were divided by a third vector (T) also drawn at random from a Normal distribution. The two vectors have nothing in common, they should exhibit no correlation as in the left plot, and yet they exhibit a correlation coefficent of > 0.59 when divided by the third vector as shown in the right plot. See the introductory section of the Supplementary Information of Lovell [-@Lovell:2015] for a more complete description of this phenomenon.

high throughput sequencing experimental designs are *always* sub-compositions. Inspection of papers in the literature provide many examples. For example, in the 16S rRNA gene sequencing literature it is common practice to discard rare OTU species prior to analysis and to re-normalize by dividing the counts for the remaining OTUs by the new sample sum. It is also common to use only one or a few taxonomic groupings to determine differences between experimental conditions. In the case of RNA-seq only the fraction of RNA of interest is sequenced, usually mRNA but other sub-fractions such as miRNA may be sequenced. All of these practices expose the investigator to the problem of non-coherence between sub-compositions. We must use compositionally-appropriate measures of correlation—more formally, we are attempting to find features that are compositionally associated. Compositional association as a more restricted measure of correlation and is explained more completely in the chapter on data transformations.

To summarize, compositional data has the following pathologies:

- The negative correlation bias means that any negative correlation observed in compositional data must be treated as suspect because it could arise simply because a different feature (or features) changed their abundance. There is currently no theoretically valid approach to identify true negative correlations in compositional data [@egozcue:AJS].
- The spurious correlation problem means that we can observe apparent postive correlations simply by chance. I describe recent work that shows that spurious correlation is tractable.

- The sub-compositional incoherence of correlation is perhaps the most insidious property, but also the easiest to recognize. Here the correlation depends on the *exact* set of features present in the dataset. If the observed correlations change when the data are subset, then sub-compositional incoherence is in play.

Thus, one major reason to use compositional data methods is that you are more likely to report robust results, and the later practical chapters demonstrate the robustness of a compositional data analysis.

Practically speaking the negative correlation bias, the occurrence of spurious correlation, and the problem of sub-compositional incoherence means that *every microbial correlation network that has ever been published is suspect*, as is *every gene co-occurrence or co-expression network* unless compositionally appropriate compositional association metric was used [@Lovell:2015;@erb:2016;@Quinn206425]. These approaches themselves have limitations and as originally constituted cannot deal with sparse data. However, recasting the data from count compositions to probability distributions allows these methods to be adapted to sparse data with some success [@bian:2017;@Quinn206425].

So can I analyze compositional data? How?

Much of the high throughput sequencing analysis literature seems to assume that data derived from high throughput sequencing are in some way unique, and that purpose-built tools must be used. However, there is nothing special about high-throughput sequencing data from the point of view of the analysis. Fortunately, the analysis of compositional datasets has a well-developed methodology [@pawlowsky2015modeling;@van2013], much of which was worked out in the geological sciences.

Atichison [-@Aitchison:1986], Pawlsky-Glahn [-@Pawlowsky-Glahn:2006], and Egozcue [-@egozcue2005], have done much work to develop rigorous approaches to analyze compositional data [@pawlowsky2011compositional]. The essential step is to reduce the data to ratios between the D. This step does not move the data from the Simplex but does transform the data on the Simplex such that the distances between the ratios of the features are linear. The investigator must keep in mind that the distances are between ratios between features, not between counts of features (re-read this several times to wrap your head around it). Several transformations are in common use, but the one I believe is most applicable to HTS data is the centred log-ratio transformation or clr. The clr, and other common data transformations are explained in the next Chapter—I bet you can't wait!

Basic idea of CoDa analysis

We want to answer the question: Have the abundances of the taxa changed? But we cannot, because as we have seen the actual abundances are not available. Instead we have relative abundances, that is the ratios between the features, and the following is a simple example.

We sequence, and have a total count of about 100 (it is a second generation GregSeq machine!)

So we get: $A_s = [71, 7, 4, 18], B_s = [1, 25, 12, 62]$

Note that these values appear to be very different between the groups. However, if we take one feature as a reference, say feature 4, and determine a ratio, i.d.:

$A_r = [74/18, 7/18, 4/18] = [4.1, 0.39, 0.22]$

$B_r = [1/62, 25/62, 12/62] = [0.02, 0.40, 0.20]$

Here we can see that if we assume one feature is constant (feature 4), then the last two are seen to be very similar in abundance relative to feature 4. Now we can infer that the majority of change is in the first feature assuming feature 4 is invariant. We cannot compare the last feature because it is assumed to be constant, that is, the assumed change in the last feature is 0. Taking the ratios, actually the logarithm of the ratios is the key concept in compositional data analysis, and further rationale is outlined in the following chapters.

Data transforms in high throughput sequencing

This chapter introduces data transformations and distance (or dissimilarity) metrics that are prevalent in the ecological literature, and that have been extensively used in analyzing high throughput sequencing datasets. It is not intended to be a comprehensive analysis of data transformations. In some cases, only one transformation is demonstrated when several transformations are obviously related.

This chapter is rather information dense, but the lessons in it are very important to understand the limitations of data analysis whenever the data are compositional. I have tried to make it easy and intuitive, but in the end you must work through the examples as best you can. All the source code needed to explore the data are included either directly here, or in an external file when the amount of code would break the flow of the narrative.

Why transform data?

Data are transformed for a variety of reasons. The first reason, is to make the data amenable to statistical assumptions for parametric tests that require the data be normally distributed with similar standard deviations in all groups. Can we transform the data to approximate a Gaussian distribution? This mode of thinking leads to the use of square-root, arcsine or Hellinger transformations since they appear to transform the data into a distribution that can be interpreted. However, as we shall see below, none of these univariate transformations is suitable, and many of these historically common transformations have recently fallen out of favour, e.g., [@Warton:2011aa].

The logaritmic transformation is now commonly used after one of the ‘count normalization’ methods described below. As we shall see below, neither count normalized, nor log count normalized data is generally the best option. The second reason for normalization is to remove or adjust the compositional nature of the data [@Weiss:2017aa], again, as we shall see this is not possible. The assumption is that any conclusion made on the transformed data represents in some way changes in the underlying abundance of the input DNA molecules as outlined in Chapter . As we shall see, this is not possible, but using a CoDa-based approach we can determine if changes in the *relative values* of DNA molecules are altered; the question is "relative to what?".

Fundamentally, the goal of any experiment is to determine something about the environment that was sampled. After all, we are attempting to use HTS to determine something of interest about the underlying environment. Thus, we need to have some equivalence between the samples before sequencing and the samples after sequencing. The simplest case would be that there would be a linear relationship between the data that we could obtain from the environment, and the data that was actually collected by HTS.

Sequencing changes the shape of all but the most ideal data:

We have seen that high throughput sequencing as currently practiced is constrained by the capacity of the instrument. Let us revisit the toy example in Figure **??**, where we had an unconstrained random sample of counts and see how well common data transformations work to recapitulate the overall shape of the data.

the In this section I show that the practical consequence of the instrument constraint can be severe in all but the most idealized datasets. An idealized dataset is one where the total number of molecules is the same per sampling effort in all the samples taken from the the environment. An example of such an idealized would be if the samples were taken from a cell culture experiment with a treatment and control group where the treatment was expected to alter only a small number of features. This would be an example of a constrained dataset: the total number of DNA molecules in the samples in the two groups would be expected to be substantially the same except for random variation. This assumption is implicit in all differential abundance tools in use.

An example of a less than ideal experiment would be comparing the total gene content in DNA isolated from two different ecosystems.

It is more desirable to think about HTS data in a multivariate way as a 'composition' because the total count of molecules in the underlying sample (the environment) is always a confounding variable [@Loven:2012aa]. This way of thinking led to multivariate data normalizations.

Commonly used transformations are misleading

The `vegan` R package manual [@vegan:2017] has a very good description of many data transformations and I recommend it to the interested reader: note that the transforms in the `vegan` package are not compositionally appropriate, and I do not recommend them in general for the reasons outlined below.

Current practice is to examine the datasets using 'relative abundance' values, that is, the proportional abundance of the features either before or after normalization for read depth. This approach is equivalent to examining the input unconstrained data of the type seen in Figure 2 in the relative abundance sample space in the bottom right panel of the figure after normalizing the total number of reads to be approximately constant. This approach will obviously lead to incorrect assumptions in at least some cases. For example, depending upon the steps chosen to compare, the blue feature, that has constant counts in the input, will be seen to either increase or decrease in abundance. Conversely, the black feature, that is always decreasing in abundance will be seen to be constant if comparing samples 1-8.

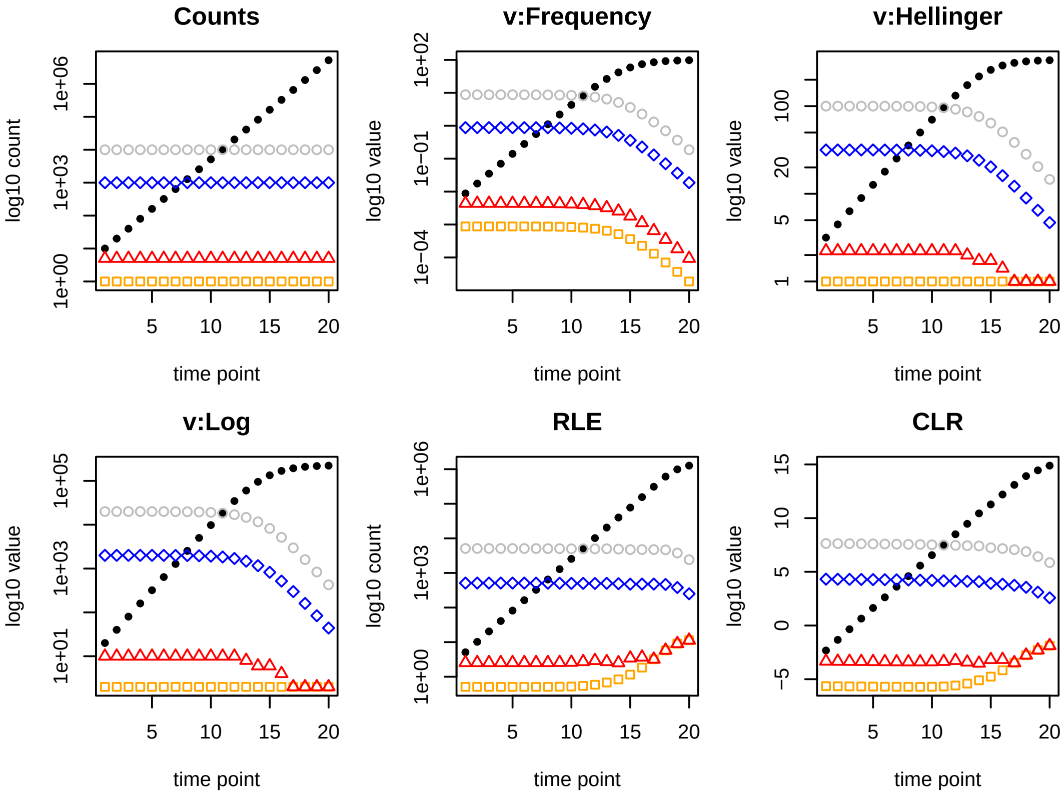


Figure 6: The effect of ecological transformations on unconstrained high throughput sequencing datasets. Data generated as in Figure 2 were normalized to a (near) constant total number of reads, converted to proportions, then transformed with five different approaches implemented in the vegan ecological analysis package. The 'Counts' panel shows the original data, the other panels begin with a 'v:' and indicate the vegan package transformation. The transformation in the RLE panel is described below.

The ecological literature offers many different transformations for such data, often as a way of making the data appear 'more normal'. Figure 6 shows the results of a few such transformations that are in the `vegan R` package.

- The frequency transform divides the each feature value by the largest feature count, and then divides the resulting values by the number of features in the sample that had non-zero counts. This is the often referred to as 'relative abundance' and is a simple proportion.
- The Hellinger transformation that takes the square root of the relative abundance

(proportion) value.

- The log transform divides each feature count in a sample by the minimum non-zero count value, then takes the logarithm of the resulting value and adds 1. Counts of 0 are assigned a value of 0 to avoid taking the logarithm of 0.

The RLE and CLR transforms are not instantiated in `vegan`, and are described below.

It is obvious that the Frequency, Hellinger, and Log transformations result in data that badly mis-represents the shape of the actual count data. All other transformations instantiated in the `vegan` package deliver data that is transformed even more extremely. The log transformation would be suitable *if* the total counts observed after sequencing was directly proportional to the total counts in the enviroment, however, we have seen that this condition cannot be met for high throughput sequencing. Thus, none of these transformations, though widely used, are suitable when analyzing high throughput sequencing data.

The RLE and CLR transforms appear to most closely recapture the shape of the original data and appear very similar. We will discuss these below.

Comparison of 'differential abundance' is problematic for compositional data [@fernandes:2013;@fernandes:2014]. Since the apparent abundance of every value depends on the apparent abundance of every other value, we can get into real difficulties if we are not careful. If we refer back to Figure 6:Counts, and we compare the relative abundances of features between sample 1 and sample 20, we observe that only the black feature has changed in absolute abundance and that the count of all other features is unchanged. Note that we would be very wron in our inferences if we use the Frequency, Hellinger or Log transforms, since we would infer that the black feature increased, but that all other features had decreased in abundance. The RLE and CLR transformations do a better job of controlling for the interdependence between features. Both transforms have the black feature increasing, but the other features increase or decrease only marginally, and apparently at random.

Other data transformations

I will introduce each of the transformations in turn, and then we will examine the effect of each transformation on the data. Let us see which, if any of the transforms fulfills the basic requirements set out above.

Notation

We use the following notation throughout.

- Column vectors contain samples $\vec{\mathbf{s}}$ and row vectors contain features $\vec{\mathbf{f}}$
- There are D features and n samples, thus the data are contained in matrix of dimension $M = D \times n$
- The j^{th} sample is denoted as $\vec{\mathbf{s}}_j$
- The i^{th} feature of all samples is denoted as $\vec{\mathbf{f}}_{i-}$

- The value for the i^{th} feature of the j^{th} sample is referred to as s_{ij} We will consider the following transformations

The proportional transformation

This simple normalization is to determine the relative abundance (rAB), or proportion, of the i^{th} feature in a sample as in Eq. 4. This normalization is also referred to as the total sum scaling (TSS) normalization. The effect is shown in panel Frequency in Figure 6.

$$rAB_i = \frac{s_i}{\sum \vec{\mathbf{s}}} \tag{4}$$

The rAB measure requires only the read count observed for the feature s_{ij} and the total read count of the sample $\sum \vec{\mathbf{s}}$. Since this measure is generally skewed, it is often log-transformed prior to analysis.

The RPKM and TPM transformations

A further normalization was proposed early in the RNA-seq field where the reads per kilobase per million mapped (RPKM)[@Mortazavi:2008] method was used initially to place the read counts for each feature within and between samples on a common scale.

For this we also needed to know a scaling factor K, and the length of the feature L_i; from this, the RPKM value for the i^{th} feature for each sample was calculated as in Eq. 5.

$$RPKM_i = \frac{K \cdot s_i}{\sum \vec{\mathbf{s}} \cdot L_i} \tag{5}$$

When the equation is placed in this form it is obvious that RPKM is simply a scaled rAB where each rAB value is divided by its length and multiplied by a constant. In compositional terms, RPKM is an unclosed perturbation of the original data; the data appear to be real numbers, but are actually proportions multiplied by a constant.

Further research suggested that RPKM was not appropriate for comparison of features between samples. The goal of RPKM was to 'count' reads per feature per cell. In the original paper the authors supplied an equivalence and an RPKM value of 1 RPKM equalled one transcript in each cell in the C2C12 cell line, but in liver cells, a value of 3 RPKM equalled one transcript per cell. Thus, from the start, this normalization was unable to normalize between-condition read counts.

The transcripts per million (TPM) normalization was advocated next [@Li:2010aa]. Patcher [@Pachter:2011] showed the equivalence between RPKM and TPM, and in compositional terms TPM is simply a compositionally closed form of RPKM multiple by a constant as in Eq. 6.

$$TPM_i = \frac{RPKM_i}{\sum RPKM} \cdot K \tag{6}$$

The rAB, RPKM and TPM normalizations are thus all very similar, differing only in the scaling of individual features, and do not allow normalization between conditions unless the samples in the environment contain *exactly* the same input number of RNA molecules. These normalizations deliver proportional data, scaled or perturbed to make the data appear as if they are numerical, and not proportional. Thus, these transformations deliver data with the same properties as shown in Figure 6, except that the scale of the y axis is altered.

A related transformation is 'rarefaction' or subsampling without replacement to a defined per-sample read count. This transformation was widely used in the 16S rRNA gene sequencing field. Rarefaction to a common read count gives a composition, that is scaled such that low count features often are replaced by 0 values [@McMurdie:2014a]. For this reason, rarefaction has now been largely replaced with the median of ratios method described below.

The median of ratios count normalization

Further work found that none of these methods were appropriate, since the read count per sample continued to confound the analyses [@Loven:2012aa]. In other words, the TMM, RPKM, TPM methods *are not scale invariant*.

Thus, the scaling normalization methods were proposed [@White:2009;@Robinson:2010a], reviewed in [@Dillies:2013]. There are several scaling normalizations, but all operate on the common assumption that by normalizing all counts in a sample to a per-sample midpoint value the normalization can impute, or at least approximate, the *number* of each feature in the underlying environment. The approaches differ largely in how the midpoint is determined. The median of ratios method called the Relative Log Expression (RLE) is instantiated in DESeq2 (and others), the trimmed mean of M values (TMM) method is used by edgeR (and others), and the Cumulative Sum Scaling (CSS) method is used by metaGenomeSeq [@White:2009] among others. The RLE method will be demonstrated and used, but the TMM and CSS methods give substantially similar results, and use the same basic logic since sample values are linearly scaled by a per-sample feature-wise midpoint; the differences are largely how those midpoints are chosen.

The RLE method calculates the ratio of the features to the geometric mean, $\mathrm{g}\vec{\mathbf{f}}_{i-}$, of each feature across all samples, and then takes as the normalization factor the median ratio per sample as the scaling factor. Each feature is then divided by the scaling factor to place each sample on an 'equivalent' count scale. The idea is that the RLE normalization 'opens' the data from being compositional to being scaled counts. It is impossible to open the data, and while the scaled counts may have some useful properties, we see below that removing compositional constraints are not among them.

The multi-step normalization RLE normalization attempts to normalize for sequencing depth thus 'opening' the data, and proceeeds as in the multistep Eq. 7. Here we start with two sample vectors $\vec{\mathbf{s}}_1$ and $\vec{\mathbf{s}}_2$, and calculate a vector of geometric means of the features $\vec{\mathbf{g}}$. Ratio vectors, $\vec{\mathbf{r}}_j$ are calculated by dividing the sample vectors by the geometric mean vector, and the median of the ratio vectors is determined. Finally, the sample vectors are divided by the median of the ratio vector for each sample.

$$\begin{aligned}\vec{\mathbf{g}} &= \mathrm{g}\vec{\mathbf{f}}i- \\ \vec{\mathbf{r}}_j &= \vec{\mathbf{s}}_j/\vec{\mathbf{g}} \\ \vec{\mathbf{d}}_j &= \vec{\mathbf{s}}_j/Md(\vec{\mathbf{r}}_j)\end{aligned} \tag{7}$$

A sample calculation is given in Table 3, and we can see that the median ratio for each sample $\vec{\mathbf{r}}_j$ samples may be different in each sample, and that the particular feature that is the median may itself be different, the median feature is in boldface in the table. Thus, by construction the feature values in each sample can be scaled by different amounts in each sample.

The RLE normalization has the attractive property that it approaches the shape of the underlying count data when the dataset is relatively well behaved. In this way it is similar to the CLR tranform. We can see this in panel RLE in Figure 6. Here, only the extreme samples at points 15-20 diverge strongly from the values observed in the underlying count data. The RLE normalization is now widely used in both the 16S rRNA gene sequencing field [REF] and in the RNA-seq field [REF]. However, we can see that the normalization fails at the margin without warning. Thus, we can never be sure if we are comparing values correctly. An additional issue is that the RLE normalization is not compositionally appropriate, and even though it (nearly) recapitulates the overall shape of the data, it does not recapitulate the *relationships* between the data across samples. Nevertheless, the RLE transformation may be somewhat useful in ideal, or nearly ideal datasets if interpreted carefully.

Table 2: Example calculation of RLE normalization

Feature	$\vec{\mathbf{s}}_1$	$\vec{\mathbf{s}}_2$	$\vec{\mathbf{g}}$	$\vec{\mathbf{r}}_1$	$\vec{\mathbf{r}}_2$	$\vec{\mathbf{d}}_1$	$\vec{\mathbf{d}}_2$
F1	1500	1000	1224.7	1.22	**0.81**	1219.5	1234.6
F2	25	15	19.4	1.29	0.77	20.3	18.5
F3	1000	500	707.1	1.41	0.71	813.0	617.3
F4	75	50	61.2	**1.23**	0.82	61.0	61.7
F5	500	1500	866.0	0.58	1.73	406.5	1851.9

What we see in this example is that the basis of the comparison (the denominator chosen) can be different for each sample. Thus, while there is some similarity to the CLR tranform, the RLE is expected to be generally less stable because the CLR uses the geometric mean of a basket of features to determine the basis. Thus, it is worth pointing out that the RLE normalization (and the TMM and CSS normalizations) can substantially change our interpretation of the data. As one example, the RLE and other normaliztions obliterate our ability to determine confidence intervals for our estimates.

Table 3: Margin of error with RLE normalizations

Count	data size	RLE?	size	MOE
400	2000	No	2000	0.182 - 0.218
200	1000	No	1000	0.175 - 0.225
50	250	No	250	0.15 - 0.25
20	100	No	100	0.122 - 0.278
400	2000	Yes	472.8	0.164 - 0.236
200	1000	Yes	472.8	0.164 - 0.236
50	250	Yes	472.8	0.164 - 0.236
20	100	Yes	472.8	0.164 - 0.236

Log-ratio transformations

Aitchison [-@Aitchison:1986] introduced the concept of the log-ratio transformation.

There are three main log-ratio transformations; the additive log-ratio (alr), centred log-ratio (clr) and the isometric log-ratio (ilr) [@Aitchison:1986;@pawlowsky2015modeling].

Using the same notation as above for a sample vector $\vec{\mathbf{s}}$ of D 'counted' features (taxa, operational taxonomic units or features, genes, etc.) $\vec{\mathbf{s}} = [s_1, s_2, ...s_D]$:

The alr is the simply the elements of the sample vector divided by a presumed invariant feature, which by convention here is the last one:

$$\begin{aligned}\vec{\mathbf{x}}_{alr} = [&log(x_1/x_D), log(x_2/x_D),\\ &\dots log(x_D - 1/x_D]\end{aligned} \tag{8}$$

This is similar to the concept used in quantitative PCR, where the relative abundance of the feature of interest is divided by the relative abundance of a (presumed) constant 'housekeeping' feature. Of course there are two major drawbacks. First, that the experimentalist's knowledge of which, if any, features are invariant is necessarily incomplete. Second, is that the choice of the (presumed) invariant feature has a large effect on the result if the presumed invariant feature is not invariant, or if it is correlated with any other features in the dataset. Interestingly, an early proposal was to use the geometric mean of a number of internal controls [@Vandesompele:2002aa], leading to the next transformation.

The ALR is similar to the RLE except that the basis is chosen beforehand and may be based on a-priori information.

The centered log-ratio transformation.

The clr is performed by taking the logarithm of the the ratio between the count value for each part and the geometric mean count: i.e., for D features in sample vector $\vec{\mathbf{s}} = [s_1, s_2, s_3, \ldots s_D]$:

$$\vec{\mathbf{s}}_{clr} = [log(\frac{s_1}{g\vec{\mathbf{s}}}), log(\frac{s_2}{g\vec{\mathbf{s}}}) \ldots log(\frac{s_D}{g\vec{\mathbf{s}}})] \tag{9}$$

where $g\vec{\mathbf{s}} = \sqrt[D]{x_1 \cdot x_2 \cdot ... \cdot x_D}$, the geometric mean of $\vec{\mathbf{x}}$.

The clr transformation is formally equivalent to a matrix of all possible pairwise ratios, but is a more tractable form. Here we are using a basket of features for the basis, not a single feature.

The clr transform is scale invariant because the same clr values are obtained from the raw counts and from the table of counts after conversion to proportions. The clr transform is sub-compositionally dominant [@pawlowsky2015modeling]. Thus, the clr transform fulfils the basic requirements of compositional data analysis.

The clr is often criticized since it has the property that the sum of the clr vector must equal 0. This constraint causes a singular covariance matrix; i.e., the sum of the covariance matrix is always a constant [@pawlowsky2015modeling]. However the clr has the advantage of being readily interpretable, a value in the vector is its abundance *relative* to a mean value.

The ilr is the final transformation, and is a series of sequential log-ratios between two groups of features. For example, the philr transformation is the series of ratios between features partitioned along the phylogenetic tree [@Silverman:2017aa], although any other sequential binary partitioning scheme is also possible [@pawlowsky2015modeling]. The ilr transformation does not suffer the drawbacks of either the alr or clr, but does not allow for insights into relationships between single features in the dataset. Nevertheless, ilr transformations permit the full-range of multivariate tools to be used, and are recommended whenever possible.

The ilr and clr are directly comparable in a two important ways: First, the distances between samples computed using an ilr and clr transformation are equivalent. Second, the clr approaches the ilr in other respects as the number of features becomes large. In this respect, the large number of features—hundreds in the case of features, thousands in the case of genes—in a typical experiment works in our favour. Thus, while not perfect, the clr is the most widely used transformation. However, care must be taken when interpreting its outputs since single features must always be interpreted as a ratio between the feature and the denominator used for the clr transformation. The problems of using clr are apparent when some subcomposition or group of taxa is analysed for further insight since the geometric mean of the subcomposition is not necessarily equal to that of the original composition, leading to potential inconsistencies.

Log-ratio values of any type do not need to be further normalized since the total sum is a term in both the numerator and the denominator. Likewise the clr value computed from RLE transformed data will be identical to the clr value computed from the raw counts because the RLE does not change the relationships between the features; with the exception of 0 count features. Thus, the same log-ratio value will be obtained for the vector of raw read counts, or the vector of normalized read counts, or the vector of proportions calculated from the counts. Thus, log-ratios are said to be equivalence classes such that there is no information in the total count (aside from precision) [@barcelo:2001].

Attempts to 'open' the data, such as with the RLE transformation, are doomed to failure because the data cannot be moved from the simplex to Euclidian space. The total count delivered by the sequencing instrument is a function of the instrument and not the number of molecules sampled from the environment, thus the total count has no geometric meaning. If the data are collected in such a way that the total count represents the actual count in the environment, then the data are not compositional and issues regarding compositional data disappear. However, at present all sequencing platforms deliver a fixed-sum, random sample of the proportion of molecules in the environment. Note that this does not mean that the read depth is irrelevant since more reads for a sample translate into greater precision when estimating the proportions [@fernandes:2013].

Comparison of transformations

A benchmark random dataset

I now set up an even simpler random dataset, composed of only four features (T, L, R, A) and 50 random samples with mean values of 100 tigers, 10000 ladybugs, 1000 Rabbits and 5 space aliens drawn from a Normal distibution—although a random uniform distribution or any other distribution will give the qualitatively the same results. I am not attempting to mimic a distribution found in a real dataset, but instead desire to show the general properties of the transformations with a simple to understand dataset. I use the dataset to show how the most common transforms compare when calculated on simulated counts, on proportions (i.e. as relative abundances after sequencing), or RLE or CLR transformed data

```
set.seed(13)
T <- rnorm(50, mean=100, sd=25)
L <- rnorm(50, mean=10000, sd=2500)
R <- rnorm(50, mean=1000, sd=250)
A <- rnorm(50, mean=5, sd=2.5)
ran.dat <- cbind(T,L,R,A)
ran.dat[ran.dat <=0 ] <- 0.1
```

The first row in Figure 7 shows the relationships between three features in the benchmark dataset as counts. We can see that the features are randomly normally distributed and uncorrelated in the scatter plots of counts. Most tools attempt to infer something about this numerical dataset using the dataset after sequencing, which we have seen does not deliver counts, but delivers a fixed sum dataset. For a proper analysis after sequencing, the data transforms must be linearly related in some way to this underlying count data from the environment.

The second row in Figure 7 shows the same pairs of features in the same data after being converted to proportions: note that this is exactly comparable to sequencing and being

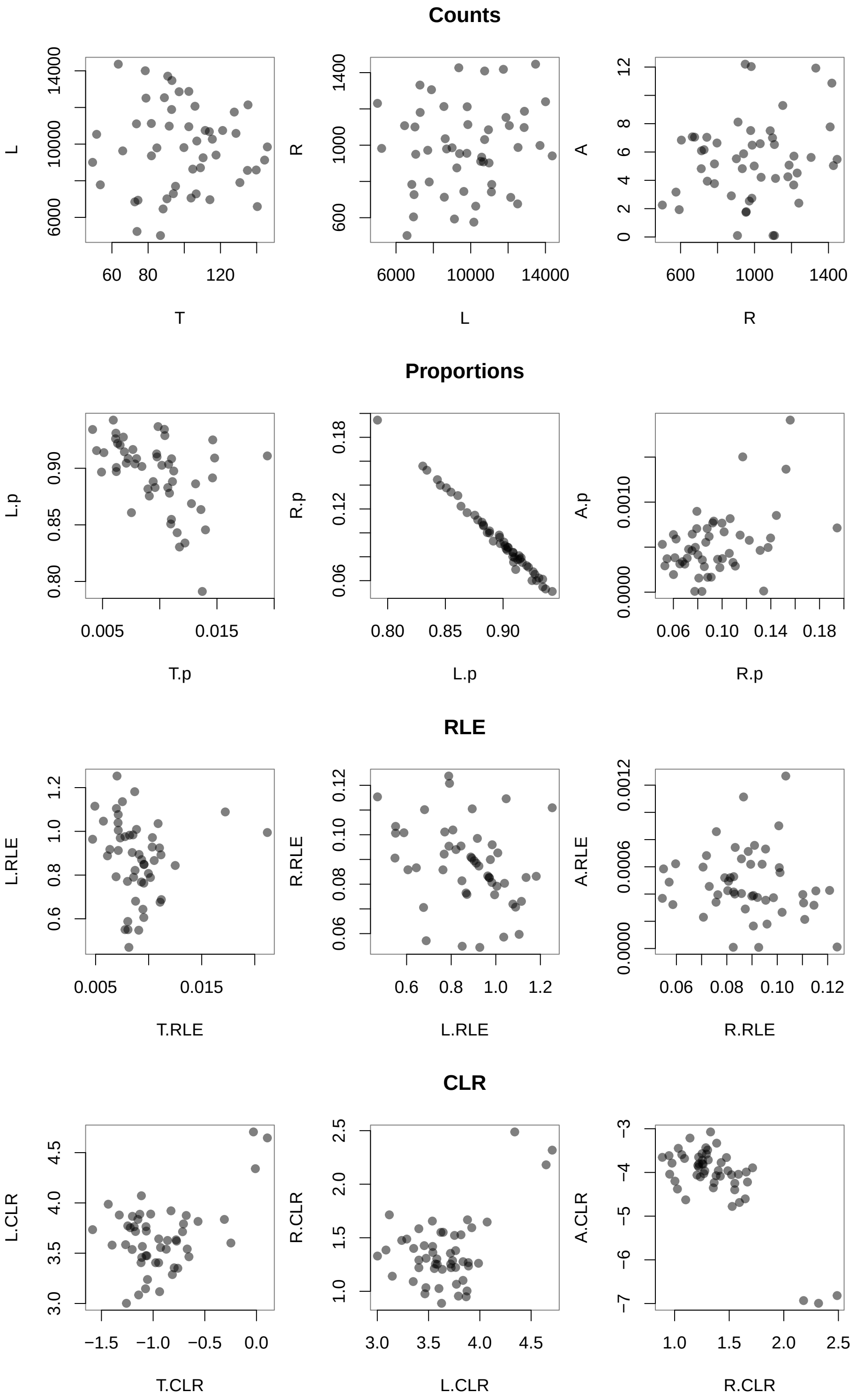


Figure 7: Scatter plots of Ladybugs (L) vs. Tigers (T), Rabbits (R) vs. Ladybugs and Aliens (A) vs. Rabbits for simulated random Normal data. Plots are shown for the actual count data, for the proportional data, and for the proportional data after the RLE or CLR transformation.

constrained by an arbitrary sum as discussed in Chapter 5. Here we see that the the proportional data have a radically different internal structure. The two most relatively abundant features, R and L, which are uncorrelated in the actual data are now almost perfectly negatively correlated as proportions, R.p vs L.p. This is because the proportional data are now not real numbers, but are instead are constrained by the arbitrary sum of 1: *the data are now compositional data.*

Recall from Figure 6 that the RLE and CLR transformations *appeared* to restore most of the underlying structure to the data. Using this simplified dataset, we can see that this was an illusion. The RLE transformation shown in row 3 of Figure 7 seems to fix the problem caused by converting the data from numbers to proportions since the points are more spread out. However, closer inspection shows that this is not the case: compare L.RLE vs T.RLE in row 3 with L vs T in row 1. It should be clear that the RLE transformation simply *spreads the points out* without restoring the actual structure of the underlying count data. This is unfortunate and misleading since the stated purpose of the RLE transformation is to recover the underlying count structure of the environmental sample after sequencing [REFS]. In the absence of a solid theoretical foundation, it is difficult to say exactly how we should interpret these RLE-transformed data. The above RLE plots were generated on the proportional dataset, however only the scale of the *RLE* axes would change if the RLE normalization was conducted on the original numerical data. Thus, conclusions derived from data that are RLE-normalized actually tell us little about the underlying counts from enviroment—*despite the pervasive use of this, and related, transformations in the biomedical literature.*

The last row in Fig 7 shows the proportional data transformed by the clr transformation. Again, we see that the transformed data are not similar to the actual count data. Thus, conclusions made on clr-transformed data also cannot be directly related to the count values of the actual dataset. So at this point we have a conundrum: no transformation on post-sequencing data recapitulates the pre-sequencing data that we want to examine.

The CLR transform contains relative information

We are now in a position to realize that recovering the actual counts in the environment from the post-sequencing data is impossible without additional information. Some advocate the use of a spike-in of known numbers of a molecule that can be used to normalize. However, in practice one would need to include a sufficient number of these molecules so that the random sampling error was very small, thus decreasing the sequencing depth of the molecules being measured in the experiment. In addition, practical examination of spike-in experiments indicates that there is considerable variation in the spike-in molecules that is attributed to batch effects [BARTON], making their use suspect.

Two transforms, the RLE and clr transforms come closest to recapitulating the overall shape of the univariate data as shown in Figure 6, and so appear to be promising. However, both transformations (and all others) fail to recapitulate the multivariate nature of the data as shown in Figure 7. There is a way forward as long as we are willing to change our point of view from absolute numbers to relative values.

Interestingly, both the RLE transform and the clr transform are based on ratios. These tranforms share the insight that we need to examine the abundance of a feature relative to the abundance of some other feature or group of features. The RLE transform uses as the reference a median value calculated as in Equation 7. Two aspects of this normalization are troubling. First, that the midpoint feature chosen as the reference is likely different for each sample. Second, that the values are scaled and interpreted as counts, even though the transformed values are now ratios. Third, that the transformed values will be different if the features have a different scale; i.e., if the features are multiplied by a constant.

In contrast, the clr tranform has a firm theoretical foundation based on compositional data analysis [@Aitchison:1986]. Data transformed by the clr are the same whether they are counts, or proportions or are multiplied by an arbitrary constant: they are what is called 'scale invariant'. We can demonstrate that the clr tranform provides the same relative information on the actual count data by transforming the count data, or the proportion data by the clr and plotting the result as shown in Figure 8. Further, clr-transformed data are explicitly interpreted as the ratio between the count (or proportion) of a feature and the geometric mean count (or proportion) of all features. We can further modify the clr tranformation to use only those features that have particular properties (such as non-0, low variance, etc) in all samples [JIA] to avoid including features with a count of 0 in the denominator. However, in this case we must keep in mind to interpret the transformed values as the ratio between the feature and the denominator.

Distances in high throughput sequencing

Distance or dissimilarity metrics

The microbiome and transcriptome literature are replete with distance metrics, and it is common to find that a single study will use several distance metrics to report their findings. This is a problem since it suggests that practitioners are unsure of the reason to use a metric; consequently, the use of more than one metric leads to data dredging and research degrees of freedom—both of which increase the chances of finding false positives in the data to a surety.

Distance metrics can be broadly divided into those that require partitioning and those that do not. The UniFrac [@Lozupone:2011aa;@unifrac:2005] and philr [@Silverman:2017aa] both require a phylogenetic tree, making these metrics applicable only to situations where the features can be so partitioned. For example, these distances are useful when examining 16S rRNA gene sequencing experiments. We have found that the unweighted UniFrac method is unreliable, and should be used with caution [@Wong:2016aa}, a point that was made in the original UniFrac paper and subsequently forgotten. The philr metric is a drop-in replacement for the weighted UniFrac distance metric and should be used whenever possible, since `philr` is an ilr transformation of the data where the sequential binary partitions are made along the phylogenetic tree. The `philr` transformation is thus compositionally appropriate. In practice, the weighted UniFrac distance metric provides similar results to the Aitchison distance, described below, and the ilr distance calculated using the philr transform approaches the Aitchison distance when the number of features is large.

Several non-phylogenetic distances are in widespread use in the literature, and since the

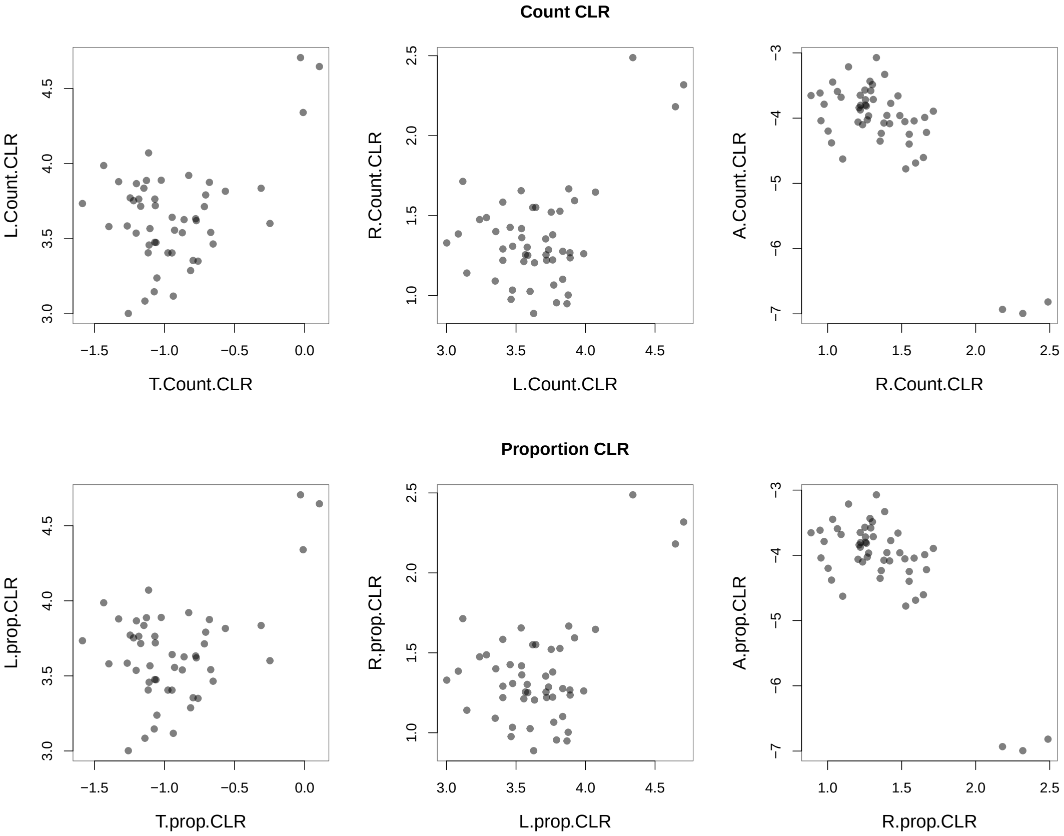


Figure 8: Plot of Ladybugs vs. Tigers, Rabbits vs. Ladybugs and Aliens vs. Rabbits for simulated random Normal data after the RLE normalization (top row). Plot of input numerical and RLE normalized data for Tigers, Ladybugs and Rabbits (bottom row)

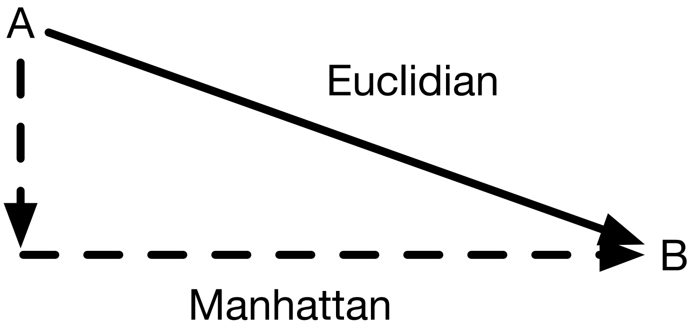


Figure 9: Two primary distance metrics are Euclidian and Manhattan. Euclidian distance is simply the straight-line distance between two points, A and B. Manhattan distance, also called city block distance, is the distance parallel to the axes of the co-ordinate system. Many other distances are in use, but these two and their derivatives are the most widespread.

phylogenetic (or partitioning) distances are difficult to apply to RNA-seq data, we will not illustrate these. Non-phylogenetic distance metrics will be discussed in turn below, and their effects on distances between a random samples illustrated.

Distances in counts and proportion}

Ideally, we use distance metrics to inform us as to something of relevance in the actual sample. That is, if we collect our data on the numbers of tigers, ladybugs, Rabbits and space aliens, what can we infer about the actual data *after sequencing*? which as we have seen, is the same as asking what can we infer after converting the data to relative abundances (proportions)?

There are two main ways to think about distances: Euclidian and Manhattan. The Euclidian distance is the straight-line distance between two points. If we have a rectangular room, the Euclidian distance between two corners would be the distance travelled by walking diagonally across the room from one corner to the other. The Manhattan distance would be the distance travelled by walking along the walls between the two corners. Obviously, the Manhattan distance will always be larger than the Euclidian distance. So how do these two simple metrics, and others derived from them, compare when calculate on numbers and on compositions?

In an ideal world when dealing with compositions, we would like a distance metric that gives us an interpretable and stable measure of distance between samples. Distances between samples should be:

- scale invariant (S): that is the distance between samples should not differ if we use proportions or percentages (or any other denominator).
- subcompositional dominant (D): that is the distance between samples that contain all the features should be equal to or greater than the distance between the samples when

one or more features are removed.
- perturbation invariant (P): that is the distance between samples should be unchanged if we translate or rotate them in space.

Martin-Fernandes [-@martin1998measures] provide a very simple test that can be used to determine if a distance metric is compositionally appropriate. We start with four samples, x1 to x4 that contain three features each, and measure the distance between the samples following a perturbation, or following feature subsetting. The perturbed samples are labeled p1 to p4, and the subset samples containing only the first two features are called s1 to s4.

```r
x1 <- c(1,2,7) * 0.1
x2 <- c(2,1,7) * 0.1
x3 <- c(3,4,3) * 0.1
x4 <- c(4,3,3) * 0.1

x <- rbind(x1,x2,x3,x4)

s <- apply(x[,1:2], 1, function(x) x/sum(x))
s <- rbind(s,c(0,0,0,0))

x.p <- t( t(x) * c(8,1,1) ) # perturbation, samples by row

p <- t(apply(x.p, 1, function(z) z/sum(z)))
```

This dataset is constructed so that the x1:x2 distance is greater than the x3:x4 distance [@martin1998measures]. This is a bit counterintuitive when examining the vectors since in both, the only difference is in the first two features, and so these features determine the distance. In both cases the two first features differ by 0.1, and so if we treat these as non-compositional data we would infer that the x1:x2 distance is equal to the x3:x4 distance. However, since the data are compositional we must interpret the relative values: the first feature in x1 is 0.5 that of the first feature in x2 whereas the first feature in x3 is x4 is 0.75 that of the first feature in x4. The second feature is the inverse relative difference. Thus, since the fold difference between the first two features of x1 and x2 are larger than the fold difference between the first two features of x3 and x4, the distances between the pairs of samples must be correspondingly different. When dealing with the subcomposition, the distances are unchanged, because the ratios between the first two features is unchanged, we have only dropped the non-informative last feature.

The relationships between the full composition and the subcomposition can be observed in Figure 10. The perturbed dataset is simply a translation of the data on the simplex and should not change the distance between samples. The ternary plot shows that our visual intuition breaks down when examining data on a simplex because the features are not linearly different in a composition. It is helpful to think that a simplex is akin to a distorted map projection where we are trying to show the relationships between the continents on a globe but projected onto a flat map.

Table 4 shows the properties of several distance metrics on the synthetic vectors. By definition, the Euclidian and Manhattan distances are not scale invariant, perturbation invariant, nor

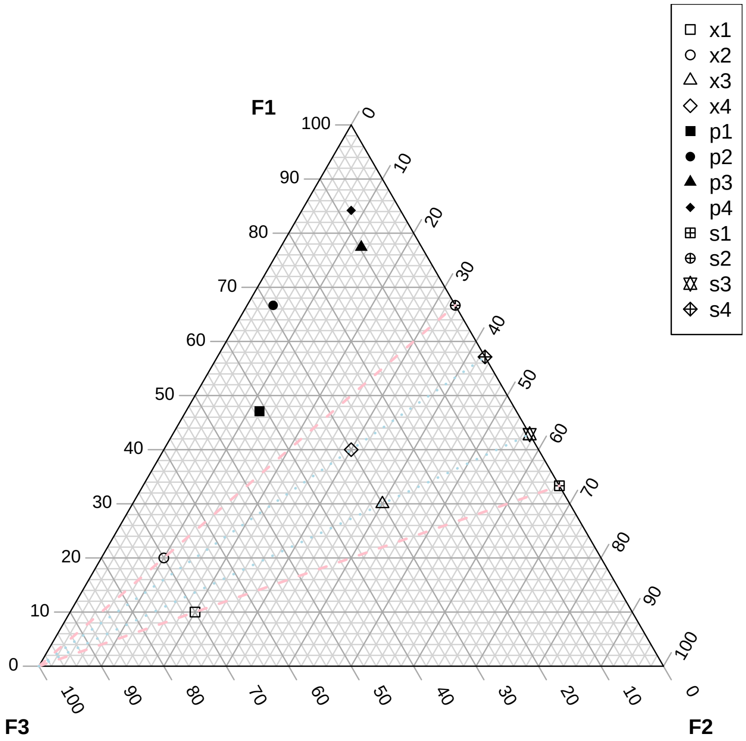


Figure 10: Ternary plot of toy data showing compositional properties. The simplex is the natural space of proportional data, or equivalently, probabilistic data, and contains one fewer dimension than does the data. The location of the four samples on the simplex are shown, along with their location after perturbation. The proportion of each feature in each sample determines the location of the feature on the simplex plot. The proportion of 1 is at the vertex with the label for each feature. The pink dashed line shows the projection of the data onto the F1-F2 proportion when feature 3 (F3) is removed. Distances on the simplex are not necessarily intuitive, since the x1, x2 distance is about twice that of the x3, x4 distance, although on the plot the distances appear similar. This is because distances are non-linear and become more distorted as the margin of the plot is approached: see [@martin1998measures] for an explanation of this.

Table 4: Distance metrics on the simplex. The x1,x2 distance should be larger than the x3,x4 distance, the distance should be the same if we change the scale (S), the perturbation should not change the distances since it is simply a translation (P), and the distance between sample subcompositions should be no larger than the full composition (D). Among the distances compared, only the Aitchison distance—the Euclidian distance of the clr values—fulfills these properties.

Metric (SDP)	d(x1,x2)	d(p1,p2)	d(s1,s2)	d(x3,x4)	d(p3,p4)	d(s3,s4)
Euclidian (—)	0.14	0.24	0.47	0.14	0.09	0.20
Manhattan (—)	0.20	0.40	0.67	0.20	0.14	0.29
Bray-Curtis (S–)	0.10	0.20	0.33	0.10	0.06	0.14
JSD (SD-)	0.13	0.15	0.13	0.08	0.06	0.08
Aitchison (SDP)	0.98	0.98	0.98	0.41	0.41	0.41

are they subompositionally dominant. These metrics should *never* be used for proportional (compositional) data. The Bray-Curtis dissimilarity (or if symmetrized Bray-Curtis distance) is scale invariant by definition since all values are scaled between 0 and 1. However, the Bray-Curtis dissimilarity is not subcompositionally dominant, nor is it perturbation invariant. Thus, the Bray-Curtis metric will be sensitive to the choice of features that are included in the analysis, and raw and normalized data are expected to give different results.

The Jensen-Shannon Distance is a symmetrized version of the Kulback-Leibler divergence metric that is widely used when comparing probability vectors [REF, REF]. This metric is both scale invariant and subcompositionally dominant. Thus, the JSD would be expected to give results consistent with the whole when used on subsets of the data. However, the JSD metric is not perturbation invariant, and so will not give the same results on the raw and transformed data.

The Aitchison distance is the Euclidian distance calculated on the log-ratio transformed data; here we use the clr transform. The Aitchison distance fulfills all properties and so is expected to give consistent results when a dataset is subsetted, when the dataset is scaled, or when the dataset is transformed. Thus, this distance metric should be used whenever possible. The utility of the distance metrics are illustrated more fully below.

Graphical demonstration of distance pathologies

I return now to the simple the Ladybugs (L), Tigers (T), Rabbits (R), and Aliens (A) dataset and examine the distance between samples using different metrics. Recall that we desire to find a distance metric that when used on the compositional data obtained after sequencing tells us something about the count data from the environment before sequencing. Thus, we should obtain a linear relationship between the count and compositional data if the distance metric is generally useful, and we compare the counts to both simple proportions and proportions after the RLE transformation for all distance metrics.

We first examine the simple Euclidian and Manhattan distances.

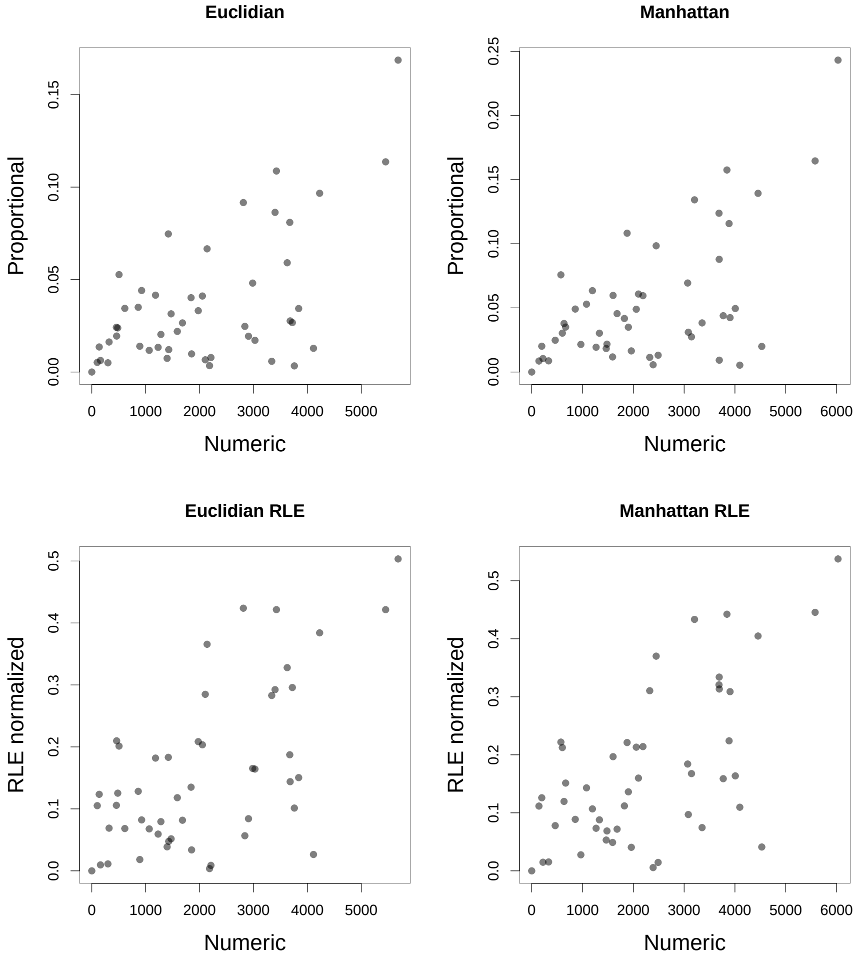


Figure 11: Scatter plot of the distances computed on numeric, proportional and RLE-normalized data. The X-axis has the distances computed on the original numeric data, and the Y-axis has the distances computed on the same data converted to proportions, or when the proportions are count-normalized using the RLE normalization. The distances for the count data and the proportions or the RLE normalized data are obviously not linearly related.

The Euclidian and Manhattan distances are generally correlated, but not identical, when comparing distances in the original numeric data, or when the data are converted to proportions. However, the distances between samples are very different when comparing the numerical and proportional data. This tells us that the inferences we make from sequencing data can not translate to inferences about the actual abundances of features in the environment, but only to their relative abundances after sequencing. So which distance metric should we use for proportional data? It turns out that neither are suitable because these distance metrics assume linear differences between features, and this is not true in proportional data [@Aitchison:1986].

Data normalizations are often touted as removing the compositionality of the data. We shall see that this is not true, and inappropriate data transformations confound, rather than providing clarity.

Plotting three of the possible combinations, we can see that the features are essentially uncorrelated with each other and each sample is a random distances from any other. Any inference we make from transformations of this data must be relatable to this 'ground truth'. I now run through each of the transformations in turn, and illustrate the difference between the actual data, and the transformed data.

Bray-Curtis Dissimilarity

The Bray-Curtis dissimilarity is a modified Manhattan distance normalized to range between 0 and 1, thus the Manhattan distance and the Bray-Curtis (BC) distances are essentially linearly related changing only the scale of the measure. One quirk of the BC dissimilarity is that it cannot be calculated if any of the values in the matrix are less than, making it incompatible with logarithmic or log-ratio transformed data.

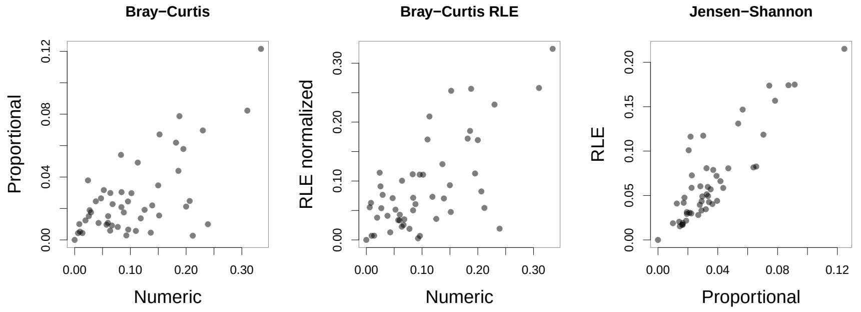


Figure 12: Scatter plot of Bray-Curtis dissimilarities, or Jensen-Shannon divergence of numerical vs. proportional and RLE normalized data.

Distances of clr transformed data

The blah blah blah

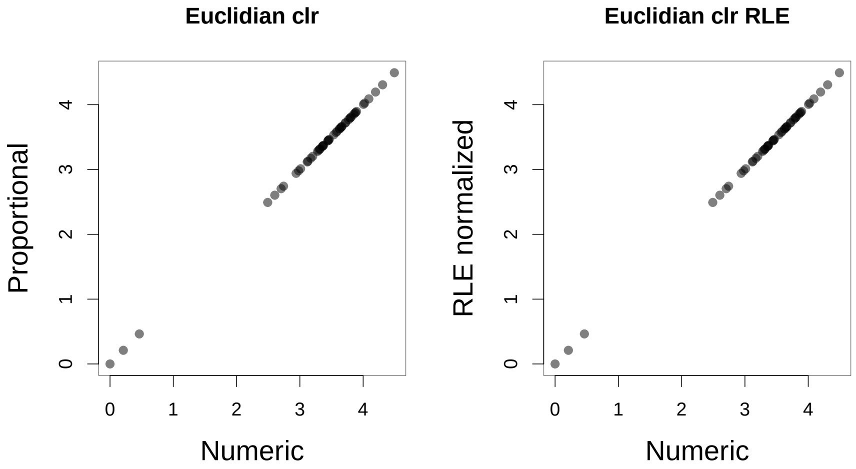


Figure 13: Scatter plot of Euclidian distances of the clr transformed numeric or proportional data. As above, RLE indicates proportional data scaled by the RLE normalization.

```
plot_ly(x=ran.dat[,"L"], y=ran.dat[,"T"], z=ran.dat[,"R"])
plot_ly(x=ran.dat.prop[,"L"], y=ran.dat.prop[,"T"], z=ran.dat.prop[,"R"])
plot_ly(x=ran.dat.RLE[,"L"], y=ran.dat.RLE[,"T"], z=ran.dat.RLE[,"R"])
```

Exploring compositional data: the compositional biplot

> This example was first presented as part of the CoDa microbiome tutorial in Barcelona, Spain in April 2018 at the NGS'18 conference. It has been modified for clarity and completeness for this book.

When analyzing and interpreting compositional data, it is important to remember that we are examining the variance in the ratios of the underlying data, and not directly examining abundance. The first tool that we will use is the compositional biplot. This is generated by the following set of steps:

1. remove essential 0 values (0s that are in all samples. i.e. nondetects)
2. perform any additional filtering (sparsity, minimal abundance, minimum sample count, etc)
3. adjust remaining 0 values with the zCompositions package
4. perform the clr transform on the data
5. conduct a singular value decomposition using prcomp
6. display the results in a principle component plot

Let us see how this works in principle. We will make a sample dataset that has 30 samples and only nine features. Samples will be in two groups. The first group of 20 (1-20) will differ from the last group of 10 (21-30). Feature A will be more abundant in the first 20 samples, and less abundant in the last 10 , features B, C and D will be the opposite. Features C and D will be simple transforms of feature B such that feature C has about a 5000-fold greater difference between groups than feature B, but the same relative variance. Feature D will have the square of the variance of feature B. The remaining features will be highly variable, but at random. For simplicity we will use a random-normal distribution, but any other distribution would work as well.

```
set.seed(7)
# runif(n, min, max)
# rnorm(n, mean, sd)
A <- c(rnorm(20, 20, 5), rnorm(10, 10, 2.5))
B <- c(rnorm(20, 10, 2.5), rnorm(10, 20, 5))
C <- B * runif(30, 4000, 6000) # perturb B by random uniform amount
D <- B ^ runif(30, 1.4, 1.5) # power by by small amount
E <- rnorm(30, 25, 6.25)
set.seed(6)
F <- rnorm(30, 25, 6.25)
G <- rnorm(30, 100, 25)
H <- rnorm(30, 1000, 250)
I <- rnorm(30, 2000, 500)

a_i <- cbind(A,B,C,D,E,F,G,H,I)
a_i[a_i<0] <-0.01

# convert to proportions
a_i.prop <- t(apply(a_i, 1, function(x){x/sum(x)}))

# clr transform
a_i.clr <- t(apply(a_i.prop, 1, function(x){log(x) - mean(log(x))}))

# leave out feature H
a_g <- cbind(A,B,C,D,E,F,G,I)
a_g.prop <- t(apply(a_g, 1, function(x){x/sum(x)}))
a_g.clr <- t(apply(a_g.prop, 1, function(x){log(x) - mean(log(x))}))
```

The compositional biplot

The compositional biplot in Figure 14 shows both the features (variables A-I) and the distances between samples on one PCA (principle component) plot.

The features are represented by the red arrows and the red letters. The origin of the arrows is the midpoint of the data; in this case the geometric mean of the dataset since we are using the centred log-ratio tranform. The length of the arrow is proportional to the standard deviation of the feature, up to the resolution of the PCA plot. The arrow points in the direction of greater relative abundance in the dataset for the feature. The resolution of the plot is the proportion of the variance explained by the principle components plotted, here the proportion explained is 0.775. With this amount of variance we have a fairly good representation of the dataset.

By design, the standard deviation of each feature is one quarter the mean value, and so the length of all arrows should be exactly the same. Note that this is obviously not true; and this shows the limitation of such a representation. The best we can hope for is that the axis of the experiment is on one of the major components as it his here. This dataset has 9 features, and so is actually represented by an 8-dimensional simplex. The PCA plot is a rotation of that simplex such that the maximum variance possible is displayed in the first two dimensions.

While a compositional biplot can contain much information [@Ait1983; @aitchison2002biplots], in the context of a microbiome or transcriptome dataset that contains hundreds or thousands of features, the complexity can rapidly become overwhelming. Thus, I will only indicate the major observations.

1. The positions of the samples (1-30) are represented by a projection of their distance relationship on the simplex, up to the limit of the variance explained. In this case, we can see that samples 1-20 and 21-30 separate into to groups, with the possible exception of sample 27.

2. The arrows (or rays) for each feature are different lengths because they are a projection (shadow) of an 8-dimensional space onto two dimensions. Thus, the obviously shorter arrows (E, H, B, C) are projecting into other dimensions; for the sake of simplicity, assume they are projecting above or below the plane of the Figure 14. The total length of each arrow (the variance of each feature) will be the same in this example if all dimensions were measured.

3. The angle between the arrows for each feature is indicative of their Pearson's correlation, just as it is in a traditional biplot, with a smaller angle implying a greater correlation. By this measure correlated sets of features could include features G and I, or could include features B,C and D. However, correlation in compositional data is not enough because we are measuring the ratios between features, not their absolute values [@aitchison2002biplots; @Lovell:2015; @Erb134536].

4. A line drawn between the tips of the arrows for any set of features is called a link. Links that pass through more than one feature are permitted and do not change the interpretation. Short links indicate possible compositional association, long links indicate no association.

5. Features represented by any two non co-incident arrows indicate pairs of features that are likely uncorrelated since they have a long link. Note that in a compositional biplot, features represented by arrows pointing in opposite directions *may or may not* indicate anti-correlated features as shown in Figure 15 for the A:C pair, which is negatively correlated by design, and the G:F pair which appears to be negatively correlated but is actually not correlated at all. Therefore, it is dangerous to infer negative correlation in compositional data because there are many apparent sources.

6. Any two arrows with equal lengths (two features with the same relative standard deviation) will have a constant ratio relationship if their direction is similar because they will be found to have a short link; that is, $F_1 = F_2 * C$, where the two features are related by a common multiplicative constant C. We can see this clearly for the B,C pair, which have the same length but differ by a factor of about 5000 in absolute abundance while keeping their relative standard deviation the same.
7. Any two arrows with dissimilar lengths that have a small angle between them but a different length are related by an exponential relationship, and will be correlated but will not be in a constant ratio as shown in Figure 15. This can be seen because they will have a long link if their variance is very different.

Figure 15 shows the relationships between pairs of interest in the compositional biplot. The top left plot shows that features A and B are indeed negatively correlated in the dataset as designed. The top right plot shows that it is dangerous to infer negative correlation in compositional data as the F:G pair also appears to be negatively correlated but in fact are not correlated at all.

When examining correlation in these datasets, we are most interested in finding those features that have a near constant variance ratio. In the context of a scatter plot of clr values this shows as pairs where the correlation is strong, and the slope of the correlation is near unity. The bottom left panel shows the plot of the B:C pair, which has a correlation of 0.093, and a slope of best fit of 0.92, close to 1. In contrast, the B:D pair has an even higher correlation of 0.99, but the slope of best fit is 0.58, and is clearly not close to 1. Thus, in this dataset only the B:C pair is likely to be compositionally associated.

Compositions are relatively stable to subsetting

We have seen that any of the count-normalization data transformations simply rescale the proportional data, and that a log-ratio transformation is thus more appropriate for compositional data. The effect of the difference in robustness of the interpretation of the data can be appreciated by the demonstration in Figure 16, where we compare the compositional biplot of all features, and the remainder after dropping feature H, which is the third most abundant feature. Note that the effect of dropping feature H from the composition is minimal. The relationships between the remaining features and samples are nearly unchanged since feature H was not contributing to the split between the sample groups.

When examining hight throughput datasets, it is common to have thousands of features or more where the majority of the features contribute little if anything to the dataset. One common tactic to simplify such complex compositional datasets is to remove those features that contribute little to the total variance of the dataset. This will be demonstrated in the practical chapters that follow where we examine real datasets.

When analyzing real datasets with compositional biplots, it is useful to subset the features in the data in different ways and examine if the overall separation between groups changes. Common subsetting choices include increasing or decreasing the proportional threshold that a feature must achieve, minimum count or average count levels of a feature, or contribution of the feature to total variance. If subsetting does not change the conclusions, then one can

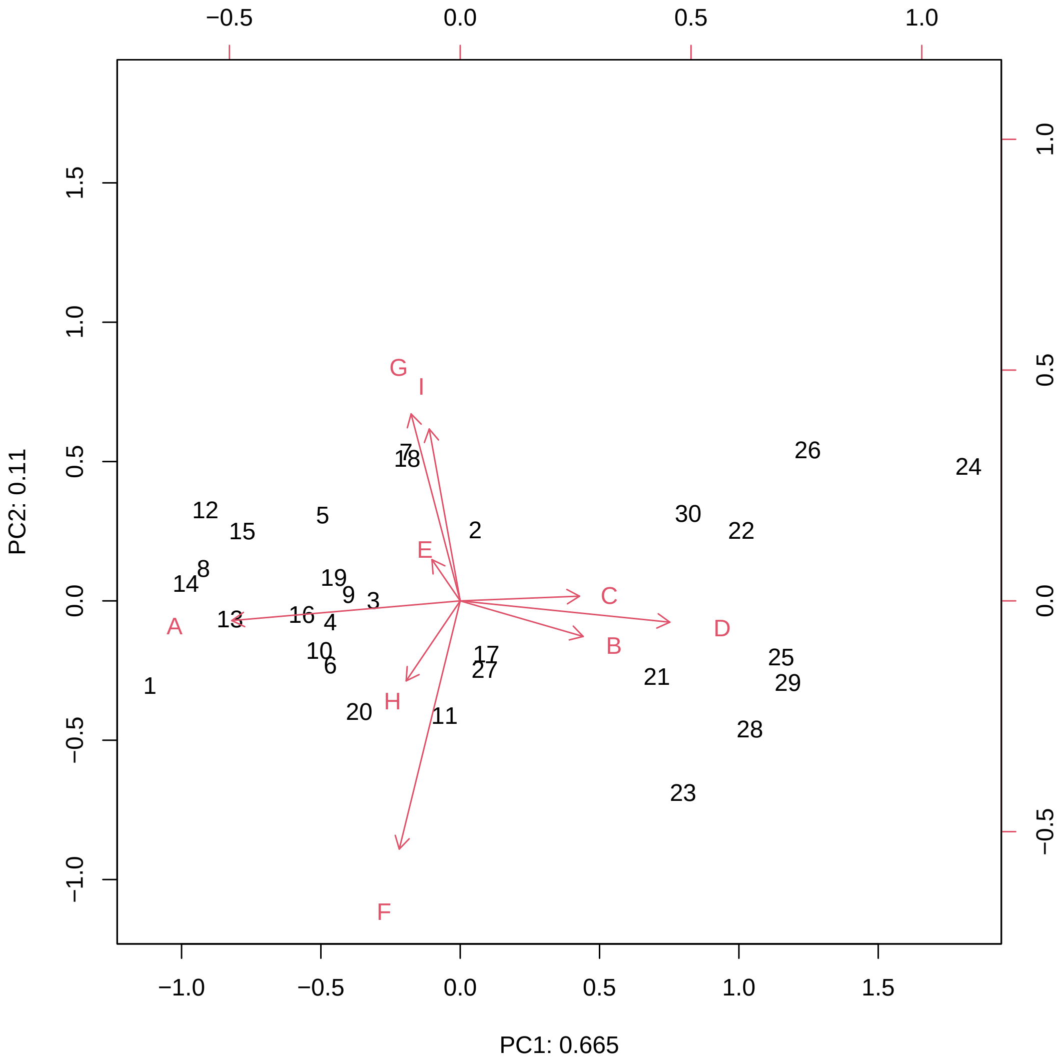


Figure 14: Compositional biplot of the features A-I. From this we can see that the absolute abundance is not represented in the biplot.

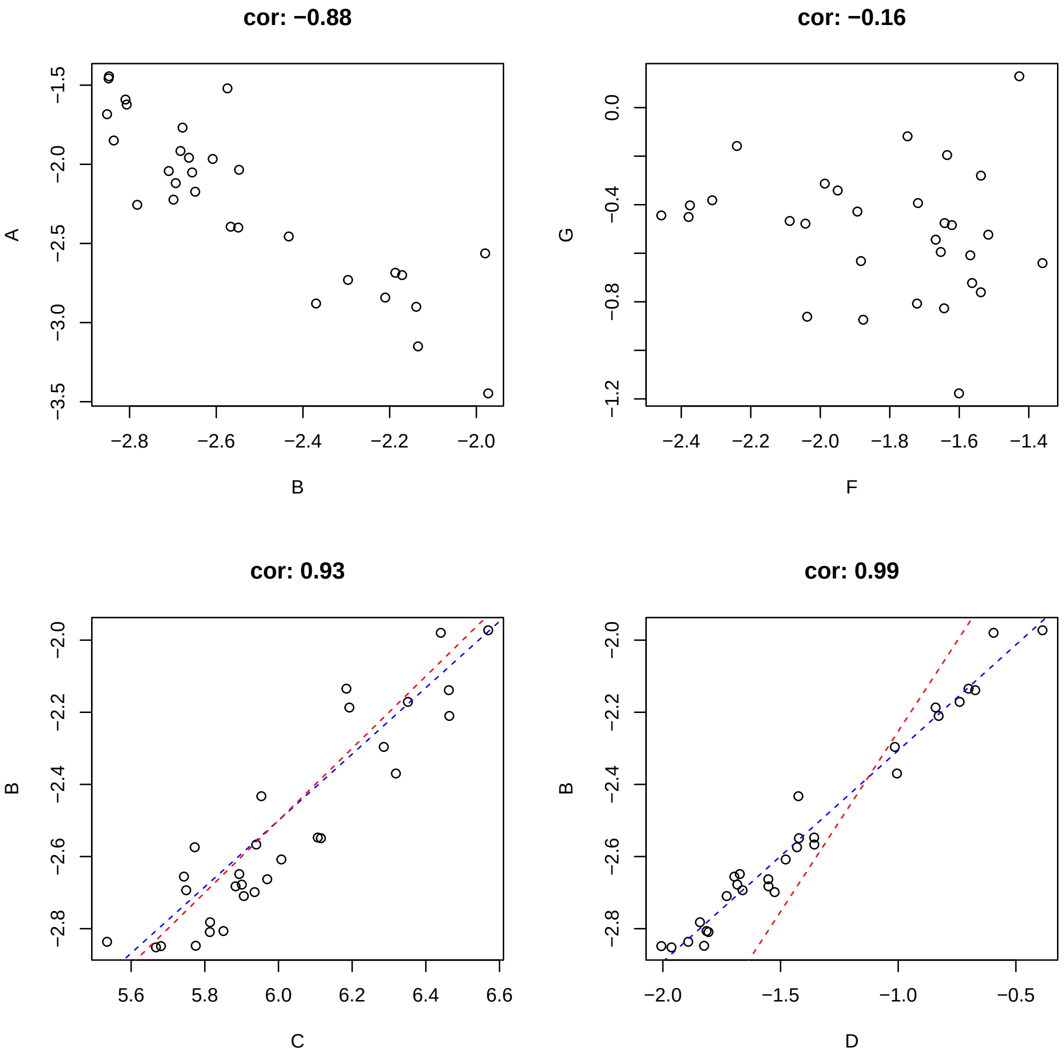


Figure 15: Counts were generated from a random normal distribution for 9 features labeled A-I. Thirty samples were generated, and features A,B,C and D were differentially abundant in the first twenty and last 10 samples. The others were of widely different abundances, but with random change between samples. Feature C is a randomly perturbed feature B, and feature D is a feature B powered by a random range between 1.4-1.5. Feature B is plotted vs features C and D as both counts and as centre log-ratio transformed values.

be sure that the arbitrary choices in data processing have little or no effect on the overall conclusions. However, if subsetting does change the separation between groups, or introduce new groupings, then one knows that the conclusions are not robust to arbitrary choices, and are not to be trusted. An example of such subsetting to show stability can be found in the supplementary information for Bian et al. [-@bian:2017].

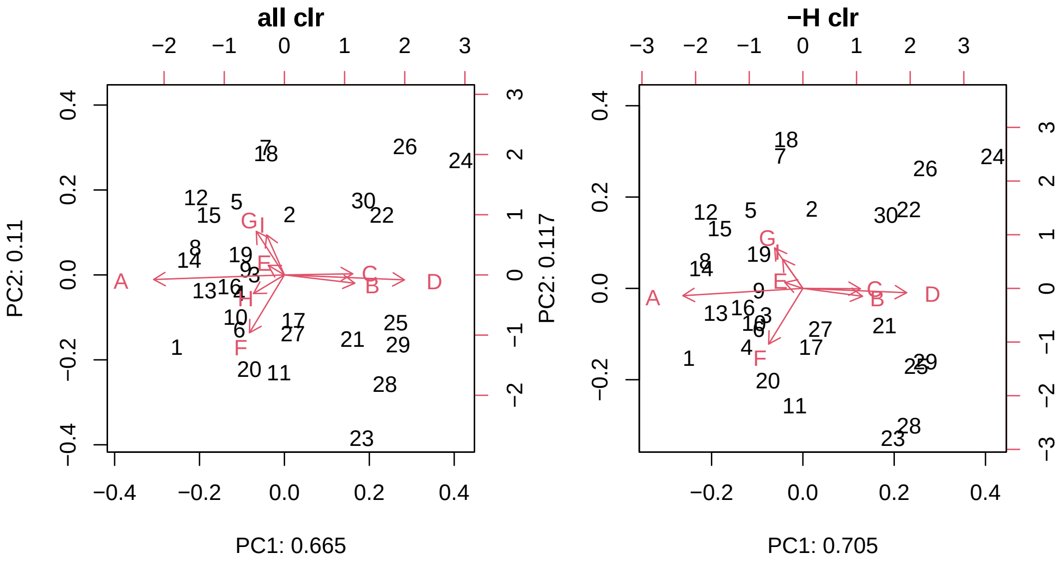


Figure 16: Compositions are largely invariant to (rational) subsetting. That is, we get essentially the same answer with all (A-I), and with a subset of the data (-H). Note that feature H has been removed on the right, but the overall conclusions drawn will be similar. In particular, the separation between groups is unchanged and the relationships between those features that remain is nearly identical.

Analyzing a yeast transcriptome dataset

> This example was first presented as part of the Biochemistry 9545 courses offered in 2016 and 2017, and at the CoDa microbiome tutorial in Barcelona, Spain in April 2018 at the NGS'18 conference. It has been modified for clarity and completeness for this book.

OK, enough pontificating, lets finally look at a real RNA-seq dataset. I would not blame the reader if they started here in the book since it is the first real dataset we look at, but I do expect that the reader will have internalized the prior chapters. We will use as an example a *Saccharomyces cerevisia* (yeast) transcriptome dataset [@Schurch:2016aa;@Gierlinski:2015aa] containing 96 samples, 48 each from wt and SNF2 knockout strain. I am using this dataset

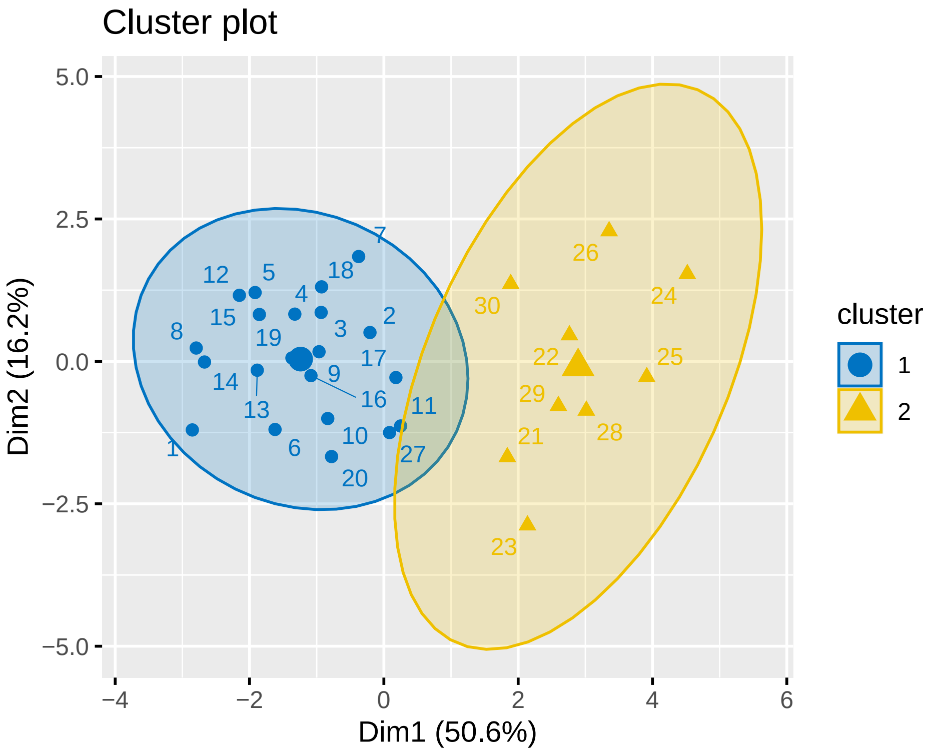


Figure 17: Samples from compositional data can easily be clustered, and fuzzy clustering is a good option for this [@fernandez:2012]. We are using the ppclust package for this [@fuzzy:2018]. Here the synthetic data are shown partitioned into two groups, that separate the two sets well. All samples cluster as expected except for sample 27, which is grouped incorrectly.

because it is in many ways ideal; there are a large number of samples, the data are from a completely sequenced model organism, the data are not particularly sparse, and it is a very simple experiment.

In any datase, the compositional biplot is the first exploratory data analysis tool that should be used. It shows, in one plot, the essence of your results. Do the samples separate into groups? which features are driving this separation? what features are irrelevant to the analysis? Does my data contain presumptive outliers?

Compositional biplots of real data appear to be complex and intimidating because of the number of samples and features, but with a little patience and practice they are easily interpretable [@aitchison2002biplots]. They are based on the variance of the ratios of the parts, and are substantially more informative that the commonly used PCoA plots that are driven largely by abundance [@Gorvitovskaia:2016aa].

```
## No. corrected values:  32498
```

Interpreting the yeast compositional biplot:

The complexity of the data makes applying the rules for interpreting compositional biplots impractical. In particular, there are so many features that there are many apparent short links, and this dataset is too complex to identify links easily. However several things stand out on the plot in Fig 18.

1) The proportion of variance explained by the first two principle components is about 31%. This is not exceptional, however, examination of the scree plot in Fig 18 shows that the first component contains substantially more information than expected at random.
2) The data samples separate clearly into two natural groups, all labelled SNF* or WT*. This separation is reassuring since the experiment was a knockout of the SNF2 gene (a knockout is a complete removal of the gene from the genome).
3) If we examine the genes, we see that the gene for SNF2, YOR290-C, is the single most variable feature in the dataset along component 1 and is at the right side just below the midline.
4) Both the SNF* and WT* sample groups appear to consist of samples that group together, and samples that are outside the main body of the groups. We can apply an outlier test to determine which if any samples are further from the middle of their group than is reasonable.
5) There are a large number of features with significant variation on component 2. These variation of these features is likely contributing to the outlier samples.

Finding outliers

We can see that there are a number of samples that appear to be outlier samples. For example, should we include SNF2.6 ,which can be found at the top left in Figure 18, in the analysis or not? One of the messages of the Barton papers [@Schurch:2016aa;@Gierlinski:2015aa] was

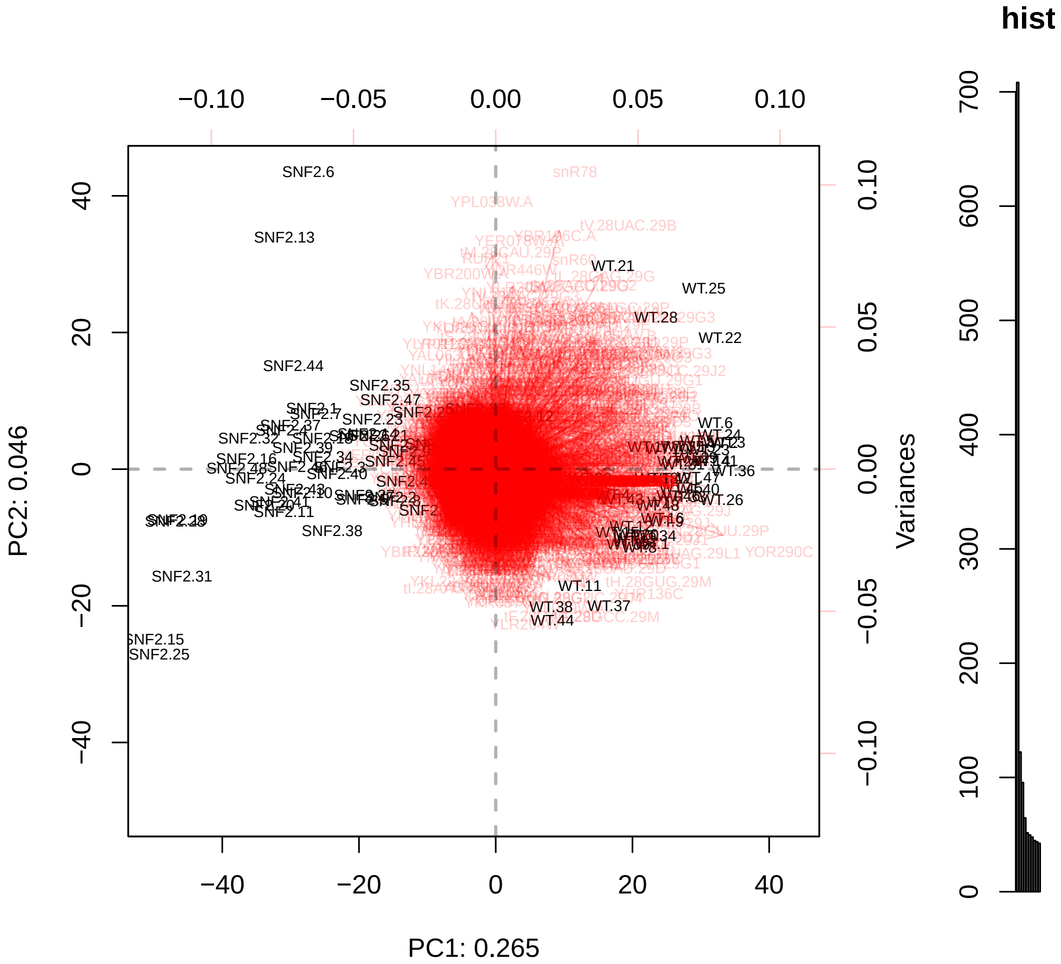


Figure 18: The compositional biplot is the workhorse tool for CoDa. This plot summarizes the entire analysis in a qualitative manner. We can see that the two groups separate well, and that component 1 has substantially more variance than does componet 2, and we can explain this experiment as a simple two part comparison with the largest variance along the axis of the comparison; i.e., along PC1.

that about 10% of samples, even carefully prepared samples such as these, can be outliers for unknown methodological reasons. I approach outliers by finding those samples that are further from the middle of the group dataset than expected. Outliers are defined as those samples that are further from the middle than the median plus twice the interquartile range of the distances of all samples in the group.

The outlier function is instantiated in the CoDaSeq R package, and the example is illustrated for the WT* group, but has been applied to both groups. The graphical intuition for how outliers are identified is shown in Figure 19. We can see from the compositional biplot of only the WT* samples, that several of them are substantially further from the rest of the samples than might be expected if the samples had homogeneous compositions that differed only by biological and technical variation.

We can use the principles of robust statistics to examine this. We expect that if the samples differ only by technical and biological variation then the distances between samples would be distributed approximately evenly around a midpoint. We can extract the samples distances from the midpoint and determine those that are greater than twice the interquartile range from the median distance and those that are within that range. A histogram of these distances can be plotted to observe if all samples excluded are egregiously different from the median (as all are here) or if the samples excluded are at or near the cut line. If the latter case is found there is not strong evidence for removing outlier samples, and this should not be performed.

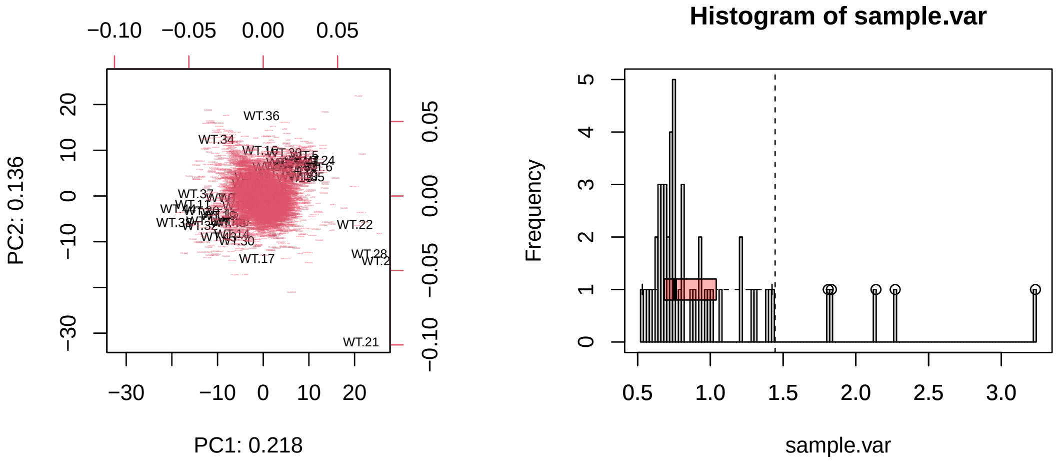
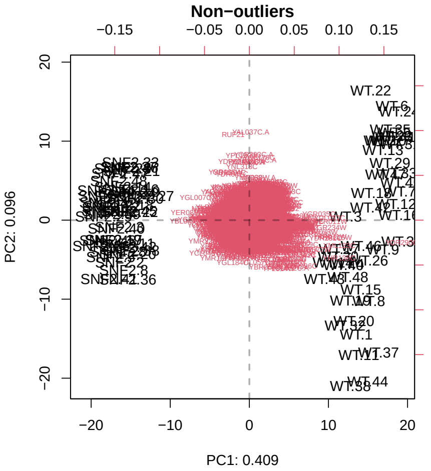


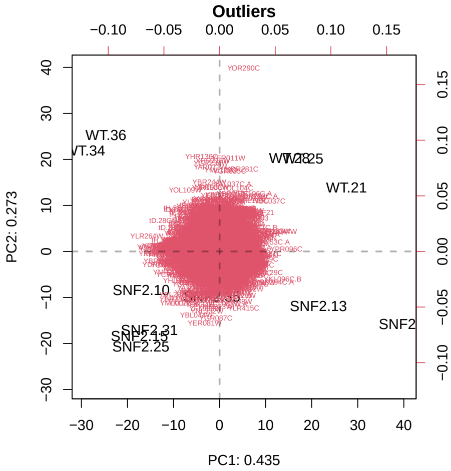
Figure 19: Identifying and removing outlier samples. The left panel shows the biplot of the WT* samples only. While the majority of the features are in a distinct ball showing little structure note that there is a set of features that projects towards the top-right of the biplot. These features are driving the separation of samples 21, 25, 28 from the rest, and less obviously 34 and 36. The right plot shows the proportion of total distance that each sample contributes to the total distance between all samples. Five samples are identified as contributing a distance that is greater than twice the interquartile range from the other samples and are excluded.

Biplot of non-outlier and outlier samples separately

It is important to check that the outlier samples are truly contributing noise or uncertainty to the system. One way we can do this is to examine a biplot of only those samples that are non-outliers.

The compositional biplot in Figure **??** :Non-outliers now explains substantially more of the variance in the data along the major axis. Furthermore, YOR290-c the knockout feature is separated from the main set of features much more obviously, suggesting that we have a more informative rotation of the data. Interestingly, we can see that the SNF2 KO group is more homogeneous than is the WT group. Almost certainly this is because the SNF2 group was clonal, and the WT group was likely grown from a frozen culture where genetic drift is known to occur.

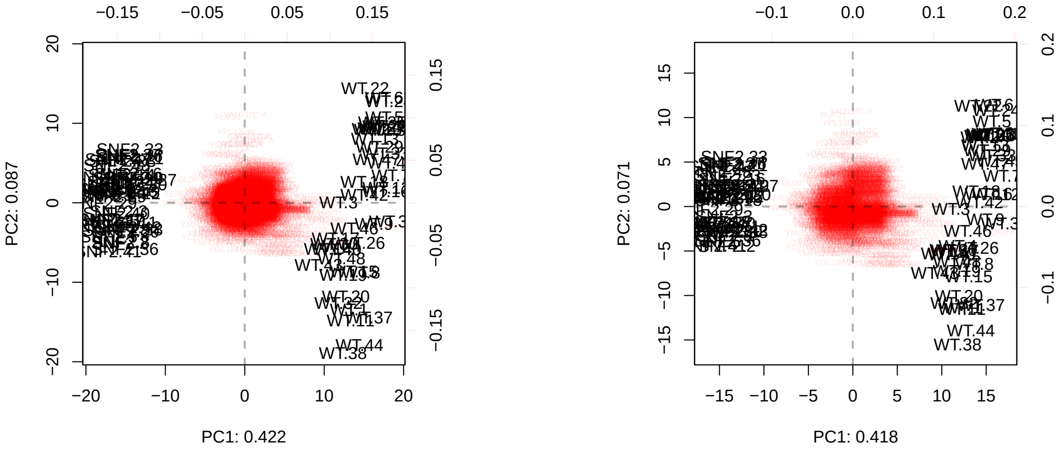




Conversly, we can also examine those samples that were deemed to be outliers. In Figure **??**:Outliers we can see that the outlier samples have the axis of the experiment on PC2, which means that there problems in the dataset that led to these samples being outliers overwhelms the actual signal in the dataset. This confirms that these samples should be removed from the dataset as they are adding either noise, or some other systematic information to the dataset.

Additional filtering does not change the conclusions

We can do additional filtering. Examining the features we see that most contribute little, if anything, to the separation between the two groups. These can be easily identified and removed by filtering out low variance features. CoDaSeq has a function to do this filtering removing those features with total variance in the dataset below the median variance. Note that we change the resolution, but that we recapitulate the dataset with only half or even a quarter of the features. We could do this iteratively as shown in the right hand panel of Figure **??**. This strategy is a good one to remove nuisance features that confound the analysis of those features that are important for the separation. However, one must be careful that the removal of is just simplifying the data and that removal is not changing the interpretation of the data. As such whenever filtering is performed the original, unfiltered analysis should be included as a reference.



FUZZY CLUSTERING

We can plot the samples according to their kmeans cluster membership. For this we are using the fuzzy clustering package ppclust [@fuzzy:2018]. There is a good introduction to fuzzy clustering in [@fernandez:2012]. Essentially, we are using a probabilistic (or possibilistic) approach to determine the number of clusters, and the cluster memberships. The vignette for this approach is at: https://cran.r-project.org/web/packages/ppclust/vignettes/fcm.html. As noted in the workshop, we get two clusters if we choose centers=2 or =3, but the SNF2 and WT groups split if we choose centers=4.

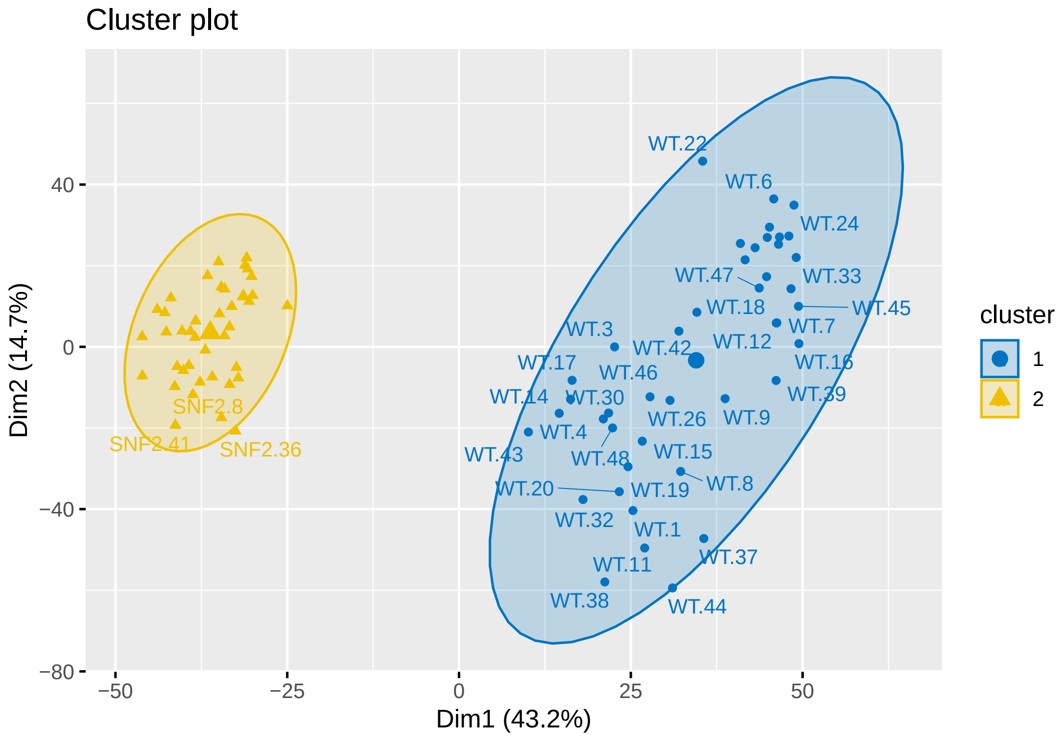
```r
library(ppclust)
library(factoextra)
library(cluster)
library(fclust)


res.fcm <- fcm(d.lv.agg.n0.clr, centers=2)
#as.data.frame(res.fcm$u)
#summary(res.fcm)

res.fcm2 <- ppclust2(res.fcm, "kmeans")

factoextra::fviz_cluster(res.fcm2, data = d.lv.agg.n0.clr,
  ellipse.type = "norm", labelsize=10,  palette = "jco",
  repel = TRUE)
```

```
## Warning: ggrepel: 49 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



############

Analyzing 16S rRNA gene sequencing experiments

The human microbiome project has initiated the large-scale culture-independent analysis of microbial communities, and transcriptome analysis has led to the study of the transcriptional response to many different disease and ecological states.

However, many studies fail to replicate earlier studies even when the same technologies and strategies are used. As just one example, a multitude of studies have examined the link between autism and the human gut microbiota. These have variously implicated x,y,. . . ,and z microbe as being linked to the condition. In a recent high-profile paper by Hsiao et al. [-@Hsiao:2013], *Bacillus fragilis* was suggested to restore the gut microbiome of a mouse autism model from 'autistic-like' to 'normal'. Examination of the dataset shows that the conclusion was likely due to chance alone [@gloor2016s;@Gloor:2016cjm]. While the autism dataset serves as a facile example, the literature are replete with other examples.

HMP oral microbiota exploration

> This example was first presented as part of the CoDa microbiome tutorial in Hinxton, UK in September 2016 at the Human-host microbe interactions conference. Dr. Jean Macklaim co-wrote and did trouble-shooting of this tutorial. It has been modified for clarity and completeness for this book.

We start the differential abundance analysis using a simple dataset derived from the original HMP oral microbiota dataset. This dataset is used because it is exceedingly sparse and so a good test of the method.

The first task is to read in the data and to generate the ALDEx2 output. The data contains 187 attached keratinized gingeva (ak) and 186 outside plaque (op) samples.

```
# The Basic ALDEx2 workflow for two conditions

# this has been modified to reduce the number of Dir monte-carlo instances
# I suggest generating sufficient Dir MCI to get at least 4000 when
# the product of the number of samples in the smallest group x DMCI

# this process can be slow, so I have pre-computed and saved the file


# read the dataset
d.subset <- read.table(paste(tutorial_data, "ak_vs_op.txt", sep=""),
    row.names=1, header=T)
# make a vector containing the two names of the conditions
# in the same order as in the column names
d.conds <- c(rep("ak", length(grep("ak", colnames(d.subset))) ),
    rep("op", length(grep("op", colnames(d.subset)))) )
# generate Monte-Carlo instances of the probability of observing each count
# given the actual read count and the observed read count.
# this returns a set of clr values, one for each MC instance, which
# constitutes a distribution of clr values
# note that the latest version of ALDEx2 requires conds explicitly
d.x <- aldex.clr(d.subset, conds=d.conds, mc.samples=22)
# calculate effect sizes for each mc instance, report the expected value
d.eff <- aldex.effect(d.x, d.conds, include.sample.summary=TRUE)
# perform parametric or non-parametric tests for difference
# report the expected value of the raw and BH-corrected P value
d.tt <- aldex.ttest(d.x, d.conds)
# concatenate everything into one file
d.all <- data.frame(d.eff,d.tt)
```

```
write.table(d.all, file=paste(tutorial_data, "ak_vs_op_aldex.txt", sep=""), sep="\t",
    quote=FALSE, col.names=NA)
```

We display the results using a number of different plots in Figure **??** to show how each plot gives a different way of exploring the data.

The mainstay that we advocate is the effect plot [@gloor:effect], that plots the constituents of normalized change, or effect size. The effect plot shows the relationship between the between group difference and the within-group dispersion for each feature. The Bland-Altman plot [@altman:1983] shows the relationship between the between group difference and the relative abundance of each feature. The volcano plot [@Cui:2003aa] shows the relationship between fold-change and the logarithm of the p-value. Finally, the effect vs. p-value plot shows the relationship between effect size and p-value.

Let's get a visual idea what is meant by the effect size as used in ALDEx2 using two significant features. OTU 3760 has an effect size of 1.04, ol 0.107, we.eBH 5.2e-31, wi.eBH 7.6e-38, and OTU 7805 is still significant but has an effect size of about 0.25, ol 0.37, we.eBH 0.0055, wi.eBH 0.0002.

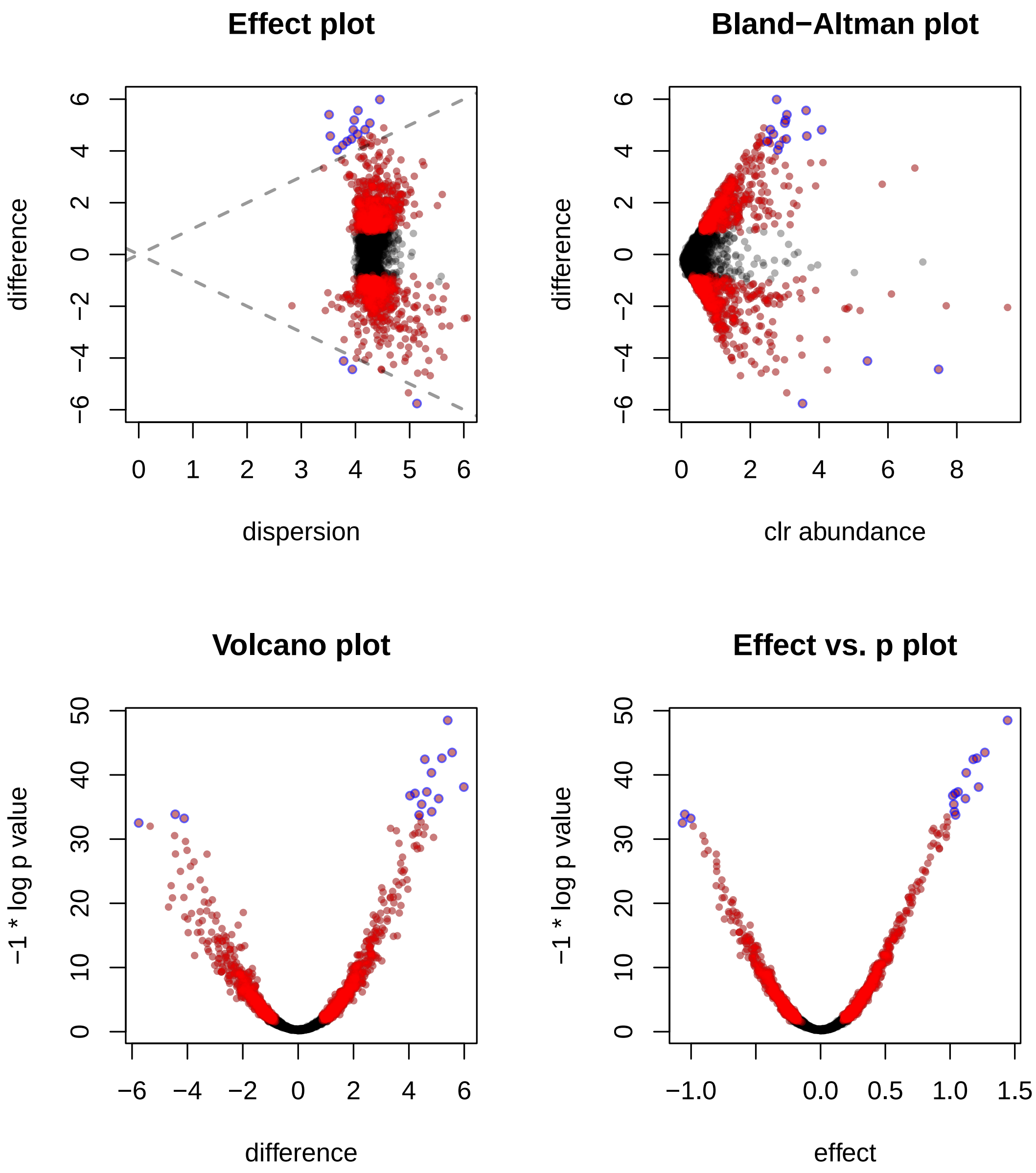


Figure 20: (#fig:hmp_aldex_plot)Plotted here are taxa with no difference between groups (grey), a statistically difference between groups (red), and with an effect larger than 1 (blue circles). These are plotted using different plots (described clockwise from top left). The effect plot [@gloor:effect] illustrates the difference between groups vs. the dispersion (variance) within groups. If the effect is greater than one (outside the grey lines), then, on average the taxa are separable by eye when plotted; roughly, they would be seen to have a greater difference than standard deviation. Effect is a more robust measure of difference than are P values, since the latter depend on sample size; large sample sizes will always give low P values [@Halsey:2015aa]. We can see this here where the large sample size means that even highly variable OTUs are significantly different. The Bland-Altman plot [@altman:1983] compares difference and abundance, and is often seen in RNA-Seq data. The Volcano plot [@Cui:2003aa] shows the association between difference and P value, and the final plot shows the association between effect and P value.

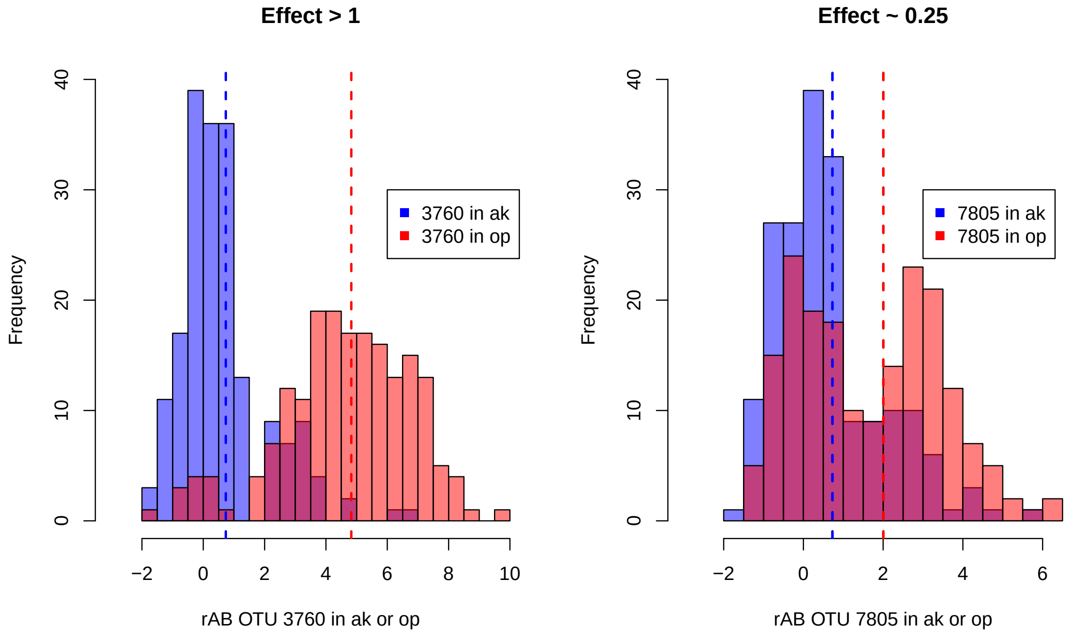


Figure 21: (#fig:hmp_aldex_effect)Histograms showing the separation between groups when choosing OTUs with large effect sizes (left), or OTUs with small effect size (right). OTUs with the largest effect are the 'most reliably different' between groups, and should be chosen over those that are 'most significantly different' whenever possible.

#References

# References

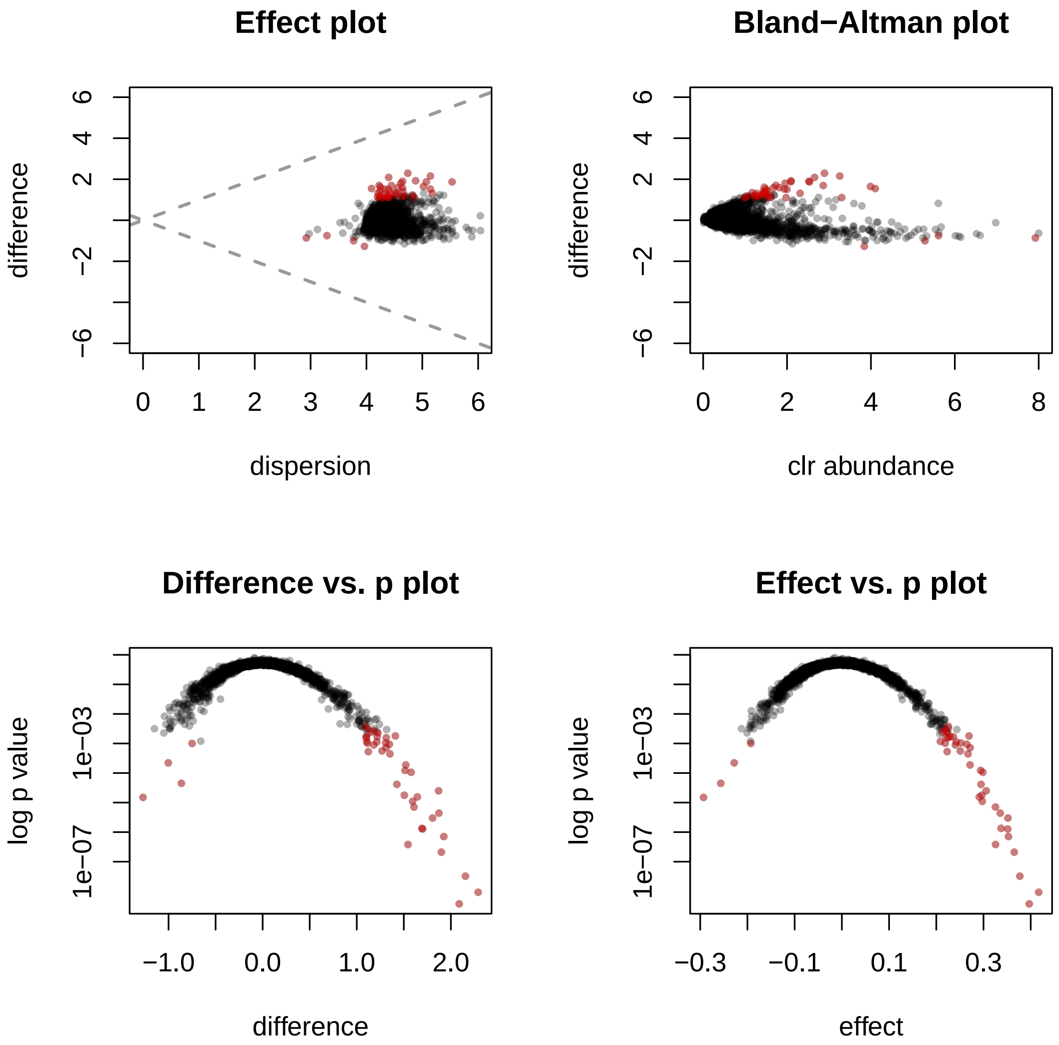


Figure 22: The same plots for the supra and subgingival plaque samples. We see that we have statistical significance, but the biological relevance is difficult to defend because of the very small effect sizes.